DOCUMENT RESUME

ED 028 947 24 SE 006 620

By-Geary, Leo C.
An Improved Algorithm for Linear Inequalities in Pattern Recognition and Switching Theory.
Pittsburgh Univ., Pa.
Spons Agency-Office of Education (DHEW), Washington, D.C. Bureau of Research.
Bureau No-BR-5-0253
Pub Date 68
Note-144p.
EDRS Price MF-$0.75 HC-$7.30
Descriptors-*Electrical Systems, *Engineering, *Mathematics, *Problem Solving

This thesis presents a new iterative algorithm for solving an n by l solution vector w, if one exists, to a set of linear inequalities, A w greater than zero which arises in pattern recognition and switching theory. The algorithm is an extension of the Ho-Kashyap algorithm, utilizing the gradient descent procedure to minimize a criterion function for a solution of the linear inequalities. The criterion function to be minimized is J(Y)=4 Sum(CoshYi) (CoshYi) where y=A w - b and b is a vector with all positive elements. This criterion function has a larger gradient than previously used and a faster rate of convergence than the Ho-Kashyap algorithm for a certain range of the initial value of b. For problems where a large number of iterations were required for the Ho-Kashyap algorithm, the proposed algorithm reduced the number of iterations by a factor of 20 to 450. The generalization of the proposed algorithm applicable to multi-class pattern classification problems is presented and a convergence proof is given. (RP)

BR-5-0253
OE-BR
PA-24

Signature

AN IMPROVED ALGORITHM FOR LINEAR

INEQUALITIES IN PATTERN RECOGNITION

AND SWITCHING THEORY

Leo C. Geary, Ph.D.

University of Pittsburgh, 1968

A new iterative algorithm is presented to solve for an n by 1 solution vector $\underline{w}$, if one exists, to a set of linear inequalities, $\underline{A}\,\underline{w} > \underline{0}$ which arises in pattern recognition and switching theory. The algorithm is an extension of the Ho-Kashyap algorithm, utilizing the gradient descent procedure to minimize a criterion function for a solution of the linear inequalities. The criterion function to be minimized is $J(\underline{y}) = 4 \sum_{i=1}^{N} (\cosh \frac{1}{2} y_i)^2$ where $\underline{y} = \underline{A}\,\underline{w} - \underline{b}$ and $\underline{b}$ is a vector with all positive elements. This criterion function has a larger gradient than previously used criterion functions. The algorithm is expressed below:

$$\underline{w}(0) = \underline{A}^{\#}\underline{b}(0),\ \underline{b}(0) > \underline{0}$$

$$\underline{y}(k) = \underline{A}\,\underline{w}(k) - \underline{b}(k)$$

$$\underline{b}(k+1) = \underline{b}(k) + p(k)\,\underline{h}(k)$$

$$\underline{h}(k) = [h_i(k)] \triangleq [\sinh y_i(k) + |\sinh y_i(k)|]$$

$$\underline{w}(k+1) = \underline{w}(k) + p(k)\,\underline{A}^{\#}\underline{h}(k)$$

where k is the iteration step and $\underline{A}^{\#}$ is the generalized inverse of the N by n pattern matrix $\underline{A}$. p(k) can be expressed as $p(k) = 1/\cosh y_{max}(k)$ with $y_{max}(k) = \underset{i}{\text{Max}}\ |y_i(k)|$ or as p(k) = Num./Den. where

$$\text{Num.} = [\underline{y}(k) + |\underline{y}(k)|]^t\ \underline{R}(k)\ [\underline{y}(k) + |\underline{y}(k)|] \text{ and}$$

$$\text{Den.} = 2[\underline{y}(k) + |\underline{y}(k)|]^t\ \underline{R}(k)[\underline{I} - \underline{A}\ \underline{A}^{\#}]\underline{R}(k)[\underline{y}(k) + |\underline{y}(k)|]$$

where $\underline{R}$ is a diagonal matrix $[r_{ii}]$ with $r_{ii} = \sinh y_i/y_i$. The algorithm also simultaneously tests for the nonexistence of a solution of the linear inequalities whenever all y_i are nonpositive with at least one y_i negative. This algorithm applies to two-category classification problems.

The algorithm has a faster rate of convergence than Ho-Kashyap algorithm for a certain range of the initial value of $\underline{b}$, $\underline{b}(0)$. A comparison has been made between the improved algorithm with p(k) = Num./Den. given above and the Ho-Kashyap algorithm with p=1. The convergence rate is greatly increased for $0.001 < b_i(0) < 0.5$ ($i=1,2,\ldots,N$) as verified by computer results of sample problems in switching theory and pattern recognition. For problems where a large number of iterations, for example, greater than twenty, were required for the Ho-Kashyap algorithm, the proposed algorithm reduced the number of iterations by a factor of 20 to 450. The total computing time was approximately reduced by a factor of three and in one case by 380 with the proposed algorithm. For problems where a small number of iterations were required by the Ho-Kashyap algorithm,

for example, less than twenty, the proposed algorithm reduced the number of iterations by as much as 30 percent.

The generalization of the proposed algorithm applicable to multi-class pattern classification problems has been presented and a convergence proof has been given. The algorithm solves for an n by R-1 solution matrix $\underline{U}$ of a set of linear inequalities $\underline{A}_j\underline{U}\ (\underline{e}_j-\underline{e}_i) > \underline{0}$, (for all $i \neq j$ and $j=1,2,\ldots,R$), where the $\underline{e}_i$'s and the R vertex vectors of a (R-1) dimensional equilateral simplex. This generalized algorithm is given in the following equations:

$$\underline{U}(0) = \underline{A}^{\#}\underline{B}(0)$$

$$\underline{Y}(k) = \underline{A}\ \underline{U}(k) - \underline{B}(k),\ \underline{Z}_j(k) = \underline{Y}_j(k)\ \underline{E}_j$$

$$\underline{B}(k+1) = \underline{B}(k) + p(k)\ \underline{H}[\underline{Y}(k)]$$

$$\underline{H}_j[\underline{Y}(k)] = [\underline{S}_j(\underline{Z}(k) + \underline{\Lambda}_j(k)]\underline{E}_j^{-1}$$

$$\underline{U}(k+1) = \underline{U}(k) + p(k)\ \underline{A}^{\#}\underline{H}[\underline{Y}(k)]$$

where again k is the iteration step,

$$\underline{S}_j(\underline{Z}(k) \triangleq [{}_{\ell}S_{jq}(\underline{Z}(k))] = [\sinh {}_{\ell}Z_{jq}(k)],\ (\ell=1,\ldots,R-1)$$

$$\underline{\Lambda}_j(k) \triangleq [{}_{\ell}\Lambda_{jq}(k)],$$

and

$${}_{\ell}\Lambda_{jq}(k) \triangleq {}_{\ell}S_{jq}[Z(k))\ \mathrm{Sgn}\ ({}_{\ell}Z_{jq}(k)).$$

p(k) is expressed as

$$p(k) = \sum_{j=1}^{R} \sum_{\ell=1}^{n_j} \{ {}_{\ell}\varepsilon_j(k) + {}_{\ell}\underline{H}_j(\underline{Y}(k))(\underline{E}_j{}^t)^{-1}\underline{R}^{-1}({}_{\ell}\underline{Z}_j(k))\underline{E}_j{}^t \; {}_{\ell}\underline{H}_j(\underline{Y}(k))$$

$$2 \sum_{q=1}^{R-1} \underline{h}_q{}^t \; (\underline{I} - \underline{A}\,\underline{A}^{\#})\underline{h}_q$$

where

$${}_{\ell}\varepsilon_j(k) \triangleq [{}_{\ell}\underline{Z}_j(k)\underline{R}({}_{\ell}\underline{Z}_j(k)) + {}_{\ell}\underline{\Lambda}_j(k)](\underline{E}_j{}^t\underline{E}_j)^{-1}\underline{R}^{-1}({}_{\ell}\underline{Z}_j(k)\underline{R}({}_{\ell}\underline{Z}_j(k)) - {}_{\ell}\underline{\Lambda}_j(k)]^t$$

$$\underline{R}({}_{\ell}\underline{Z}_j(k)) \triangleq \text{a diagonal matrix } [r_{ii}({}_{\ell}\underline{Z}_j(k))]$$

and

$$Y_{ii}({}_{\ell}\underline{Z}_j(k)) \triangleq \frac{\text{Sinh } {}_{\ell}Z_{jq}}{{}_{\ell}Z_{jq}}, \; (i=1,2,\ldots,R-1).$$

The proof of convergence of this multiclass algorithm utilizes the concept of mapping the pattern classes into vertices of the equilateral simplex.

AN IMPROVED ALGORITHM FOR LINEAR
INEQUALITIES IN PATTERN RECOGNITION
AND SWITCHING THEORY

by

Leo Charles Geary

B.S. in E.E., University of Pittsburgh, 1964

M.S. in E.E., University of Pittsburgh, 1965

Submitted to the Graduate Faculty of the
School of Engineering in partial
fulfillment of the requirements
for the degree of
Doctor
of
Philosophy

University of Pittsburgh

1968

FOREWORD

The author would like to thank his major advisor, Dr. C. C. Li, Professor of Electrical Engineering, for his invaluable guidance and support given during the preparation of this dissertation. Much of the material presented here is the result of many stimulating discussions the author had with Professor Li.

The author also would like to thank the other members of his oral examination committee, Professor T. W. Sze, Professor J. G. Castle, Professor M. H. Mickle, of the Department of Electrical Engineering, and Professor H. Wolfe of the Department of Industrial Engineering, for their helpful suggestions and recommendations.

In addition the author would like to thank the National Aeronautics and Space Administration and the University of Pittsburgh for awarding him a Predoctoral Traineeship which provided the necessary financial support for three years of graduate study. The author also would like to thank the Learning Research and Development Center at the University of Pittsburgh for the necessary financial assistance for the fourth year of graduate study and computer time.

Also, the author would like to thank Dr. W. G. Vogt and Mr. T. L. Teng for their helpful and interesting discussions. The author would also like to thank Mr. T. Brand who suggested and wrote the program for the generalized inverse of a matrix and Miss Karen Klein for her assistance in the computer program.

The author would like to thank Miss Patricia Wesesky for typing this dissertation.

Finally, this dissertation would be incomplete without mentioning the contribution of the author's wife Dottie to whom this dissertation is dedicated. Without her understanding and encouragement the author would have had a difficult assignment.

TABLE OF CONTENTS

Page

I. INTRODUCTION

A. General Background

A great amount of research for the solution of linear inequalities has been undertaken in the past ten years. One of the reasons for this research is the development of linear separation approaches to pattern recognition[(1-5,12-19)*] and threshold logic problems.[(6-11)] Both of these problems require the determination of a decision function or decision functions which, in the case of linear separation, involve a system of linear inequalities.

1. Pattern Recognition

The problem of pattern recognition requires the consideration of three fundamental aspects: namely, characterization, abstraction, and generalization.[(13)] The characterization aspect is concerned with the measurement selection and feature extraction. From the measured patterns or raw data, a set of independent variables are selected to describe the patterns under consideration. These independent variables are known as primary attributes or measurements, and are denoted by $u_1,u_2,\ldots u_d$. These attributes can be further processed to give a set of independent variables $x_1,x_2,\ldots x_r$, where

$$x_i=\phi_i(u_1,\ldots u_d), \quad i=1,2,\ldots r,$$

*Parenthetical references placed superior to the line of text refer to the bibliography.

which adequately characterize the original patterns for the purpose of classification. The vector $\underline{x}$ formed by the components $x_1, \ldots x_r$ is called the pattern vector. The abstraction aspect is the determination of the decision functions or discriminant functions so as to separate the given sample patterns according to their respective classes. This aspect is also called the training aspect. For a R-category pattern classification problem, a set of R discriminant functions, $g_j(\underline{x})$, $j = 1,2,\ldots R$, are to be determined from N sample patterns of known classification such that $g_j(\underline{x}) > g_i(\underline{x})$ for all $i \neq j$ if the pattern $\underline{x}$ is of class C_j. For a two-category classification or dichotomization problem, a single discriminant function,

$$g(\underline{x}) = g_1(\underline{x}) - g_2(\underline{x}),$$

may be used so that it separates all the sample pattern vectors into two classes. Thus the function, $g(\underline{x})$, must satisfy the following two inequalities:

$g(\underline{x}) > 0$ for all sample patterns belonging to the class C_1,

$g(\underline{x}) < 0$ for all sample patterns belonging to the class C_2.

The ability of the determined discriminant functions to recognize correctly the class of new sample patterns is considered the generalization aspect which assesses the error rate after training.

2. Switching Problems

The switching problems referred to here are a special class of pattern classification problems in which the primary attributes are the r independent switching variables of the problem, $x_1, x_2, \ldots x_r$. Each of these variables can assume only one of two values which can be represented by either 0 and 1 or -1 and 1. The pattern vectors, $\underline{x}$'s, are the vertices of a r-cube, which are 2^r in number. Every vertex of the hypercube may belong to either one of the two classes or remain unspecified with regard to its class. A Boolean function $g(\underline{x})$, known as a switching function is associated with every switching problem. It is a decision function to separate the vertices into two classes. Such a decision function can be realized by a threshold logic circuit. Thus the switching problem is essentially an abstraction problem and the techniques of linear inequalities have been applied to such switching problems.(6,10,11)

B. Ho-Kashyap Algorithm

The deterministic abstraction problem for two-category classification, as mentioned above is to determine a decision function, $g(\underline{x})$, of the pattern vector $\underline{x}$ such that

$$g(\underline{x}) > 0 \qquad \text{if } \underline{x} \text{ belongs to class } C_1$$

$$g(\underline{x}) < 0 \qquad \text{if } \underline{x} \text{ belongs to class } C_2$$

for all of the N sample pattern vectors. For the linear separation of the pattern vectors $\underline{x}$'s, $g(\underline{x})$ is a linear decision function represented by

$$g(\underline{x}) = w_1 + w_2 x_1 + \ldots + w_{r+1} x_r,$$

where the weight components $w_1, w_2, \ldots w_{r+1}$ are to be determined. For notational simplicity, let $\underline{x}$ be now redefined as an n by 1 augmented pattern vector whose first component is unity and the remaining (n-1) components are the pattern components $x_1, x_2, \ldots x_r$ mentioned previously, where $n=r+1$. The transpose of $\underline{x}$ is

$$\underline{x}^t = (1, x_1, x_2, \ldots x_r). \tag{1.1}$$

Let the transpose of the n by 1 weight vector be

$$\underline{w}^t = (w_1, w_2, \ldots w_n). \tag{1.2}$$

The discriminant function for the dichotomization problem is

$$g(\underline{x}) = \underline{x}^t \underline{w}. \tag{1.3}$$

Among the N sample or training patterns, let n_1 of them belong to class C_1 and n_2 of them to class C_2, where $n_1 + n_2 = N$. They are designated respectively by ${}_i\underline{x}_1$, $(i = 1,2,\ldots n_1)$, and ${}_i\underline{x}_2$,

($i = 1,2,\ldots n_2$), where the subscript on the right denotes the pattern class and the subscript on the left denotes the ith pattern in that class. Then the problem is to determine a weight vector $\underline{w}$ such that

$$ {}_i\underline{x}_1^t\,\underline{w} > 0 \qquad \text{for } i = 1,\ldots n_1, $$

and (1.4)

$$ {}_i\underline{x}_2^t\,\underline{w} < 0 \qquad \text{for } i = 1,\ldots n_2. $$

Ho and Kashyap have developed an iterative algorithm to solve for $\underline{w}$, which is considered one of the best available algorithms.(12) Let $\underline{A}$ be the N by n matrix of sample patterns defined below:

$$ \underline{A} = \begin{bmatrix} {}_1\underline{x}_1^t \\ {}_2\underline{x}_1^t \\ \vdots \\ {}_{n_1}\underline{x}_1^t \\ -{}_1\underline{x}_2^t \\ -{}_2\underline{x}_2^t \\ \vdots \\ -{}_{n_2}\underline{x}_2^t \end{bmatrix}. \tag{1.5} $$

Inequality (1.4) then becomes inequality (1.6):

$$\underline{Aw} > \underline{0}. \tag{1.6}$$

Let $\underline{b}$ be a N by 1 vector with all positive components and $\underline{y}$ be the N by 1 vector defined by

$$\underline{y} = \underline{Aw} - \underline{b}. \tag{1.7}$$

The Ho-Kashyap iterative algorithm for a solution of $\underline{w}$ is given by

$$\begin{cases} \underline{w}(0) = \underline{A}^{\#}\underline{b}(0), \qquad \underline{b}(0) > \underline{0} \text{ but otherwise arbitary} \\ \underline{y}(k) = \underline{Aw}(k) - \underline{b}(k) \\ \underline{w}(k+1) = \underline{w}(k) + p\underline{A}^{\#}[\underline{y}(k) + |\underline{y}(k)|], \qquad 0 < p \leq 1 \\ \underline{b}(k+1) = \underline{b}(k) + p[\underline{y}(k) + |\underline{y}(k)|] \end{cases} \tag{1.8}$$

where k denotes the iteration number and $\underline{A}^{\#}$ is the generalized inverse of $\underline{A}$.[20] The algorithm is exponentially convergent and a solution of $\underline{w}$ can be obtained in a finite number of iterations when all of the components of $\underline{y}(k)$ become positive or zero, provided that the given sample patterns are linearly separable.

The Ho-Kashyap algorithm was developed from the view point of minimizing a criterion function $J = ||\underline{Aw} - \underline{b}||^2 = ||\underline{y}||^2$. The derivation consists of the following two steps: (1) for a fixed $\underline{b} > \underline{0}$, determine a $\underline{w}$ to be a least square fit to $\underline{Aw} - \underline{b} = \underline{0}$, and (2) for a fixed $\underline{w}$, allow $\underline{b}$ to change in the direction of steepest

descent of J, subject to the constraint $\underline{b} > \underline{0}$. This algorithm has a high convergence rate for a number of pattern recognition problems. It also provides a test for nonlinear separability of the sample patterns. If the given sample patterns are not linearly separable, that is, the system $\underline{Aw} > \underline{0}$ is inconsistent, this is indicated at a certain step k^* in the iteration by $\underline{y}(k^*) \leq 0$ which is defined as all components of $\underline{y}(k^*)$ are negative or zero but with at least one non-zero component.

The generalization of the Ho-Kashyap algorithm to multi-class pattern classification has been attempted by Blaydon[15] and Fu and Wee[16] and Li, et el[23]. Experimental results have also been reported.

C. Objectives of the Dissertation

As ascertained by Devyaterikov, Propoi, and Tsypkin[18], a general recursive formula can be obtained for the system of inequalities (1.6) by minimizing a suitably chosen convex criterion function $J(\underline{y})$. In addition to the original Ho-Kashyap algorithm which uses $J(\underline{y}) = \sum_{i=1}^{N} y_i^2$, other well known non-parametric learning algorithms may also be interpreted as obtained from minimizing different criterion functions $J(\underline{y})$, for example, $J = |\underline{y}| - \underline{y}$ for perceptron's training algorithm, and $J = (|\underline{y}| - \underline{y})^2$ for the relaxation type training algorithm. Thus the solution of a system of linear inequalities can be made equivalent to a minimization problem.

With this concept as the motivation, it has been attempted to choose another criterion function J having steeper gradient than Ho-Kashyap's with a hope to further accelerate the convergence of the algorithm. Thus, the main objectives of this dissertation are: (1) to develop an improved iterative algorithm for the two-category classification problem with the choice of

$$J(\underline{y}) = 4\sum_{i=1}^{N} (\text{Cosh } 1/2\ y_i)^2,$$

and (2) to generalize this algorithm for multiclass pattern classification. The convergence proofs are given in Chapter II and Chapter V respectively. The improvement on the convergent rate has been demonstrated by a number of computer experiments on switching problems and pattern recognition problems. These experimental results are presented in Chapters III and IV.

II. AN ACCELERATED ALGORITHM OF LINEAR INEQUALITIES FOR DICHOTOMIZATION

A. Development of the Algorithm

In this chapter, an accelerated iterative algorithm will be developed for the solution of the set of linear inequalities (1.6) which is rewritten in the following equation:

$$\underline{A}\,\underline{w} > \underline{0}\ . \tag{2.1}$$

This algorithm is an improvement of the Ho-Kashyap algorithm by choosing a criterion function

$$J(\underline{y}) = 4 \sum_{i=1}^{N} \left(\cosh \frac{1}{2} y_i\right)^2 \tag{2.2}$$

to be minimized where y_i is the ith component of the N by 1 vector $\underline{y}$ defined in equation (1.7), that is,

$$\underline{y} = \underline{A}\,\underline{w} - \underline{b}\ , \quad \underline{b} > \underline{0}\ . \tag{2.3}$$

The improvement lies in an acceleration of the Ho-Kashyap algorithm caused by a steeper gradient of $J(\underline{y})$ as can be seen when a comparison is made between the two criterion functions. Let $J_{hk}(\underline{Y})$ designate the criterion function used in the Ho-Kashyap algorithm,

$$J_{hk}(\underline{y}) = ||\underline{y}||^2 = \sum_{i=1}^{N} y_i^{\,2} \, . \tag{2.4}$$

Since $J(\underline{Y})$ and $J_{hk}(\underline{y})$ reach their respective minimal when each $(\cosh \frac{1}{2} y_i)^2$ and each $y_i^{\,2}$ are respectively minimized, one can simply compare $J(y_i)$ and $J_{hk}(y_i)$, the convex functions of one variable only, where

$$J(y_i) = 4(\cosh \tfrac{1}{2} y_i)^2 \tag{2.5}$$

and

$$J_{hk}(y_i) = y_i^{\,2} \tag{2.6}$$

These two functions are illustrated in Figure 1. Taking the gradients of $J(y_i)$ and $J_{hk}(y_i)$ with respect to y_i, one obtains

$$\begin{aligned} \frac{\partial J(y_i)}{\partial y_i} &= 4 \, (\cosh \tfrac{1}{2} y_i) \, (\sinh \tfrac{1}{2} y_i) = 2 \sinh y_i \\ &= 2 \, y_i + \frac{2}{3!} \, y_i^{\,3} + \frac{2}{5!} \, y_i^{\,5} + \ldots \end{aligned} \tag{2.7}$$

and

$$\frac{\partial J_{hk}(y_i)}{\partial y_i} = 2y_i \tag{2.8}$$

It is clear that the absolute value of $\frac{\partial J(y_i)}{\partial y_i}$ is greater than the absolute value of $\frac{\partial J_{hk}(y_i)}{\partial y_i}$ everywhere except at $y_i = 0$ where they are equal. In

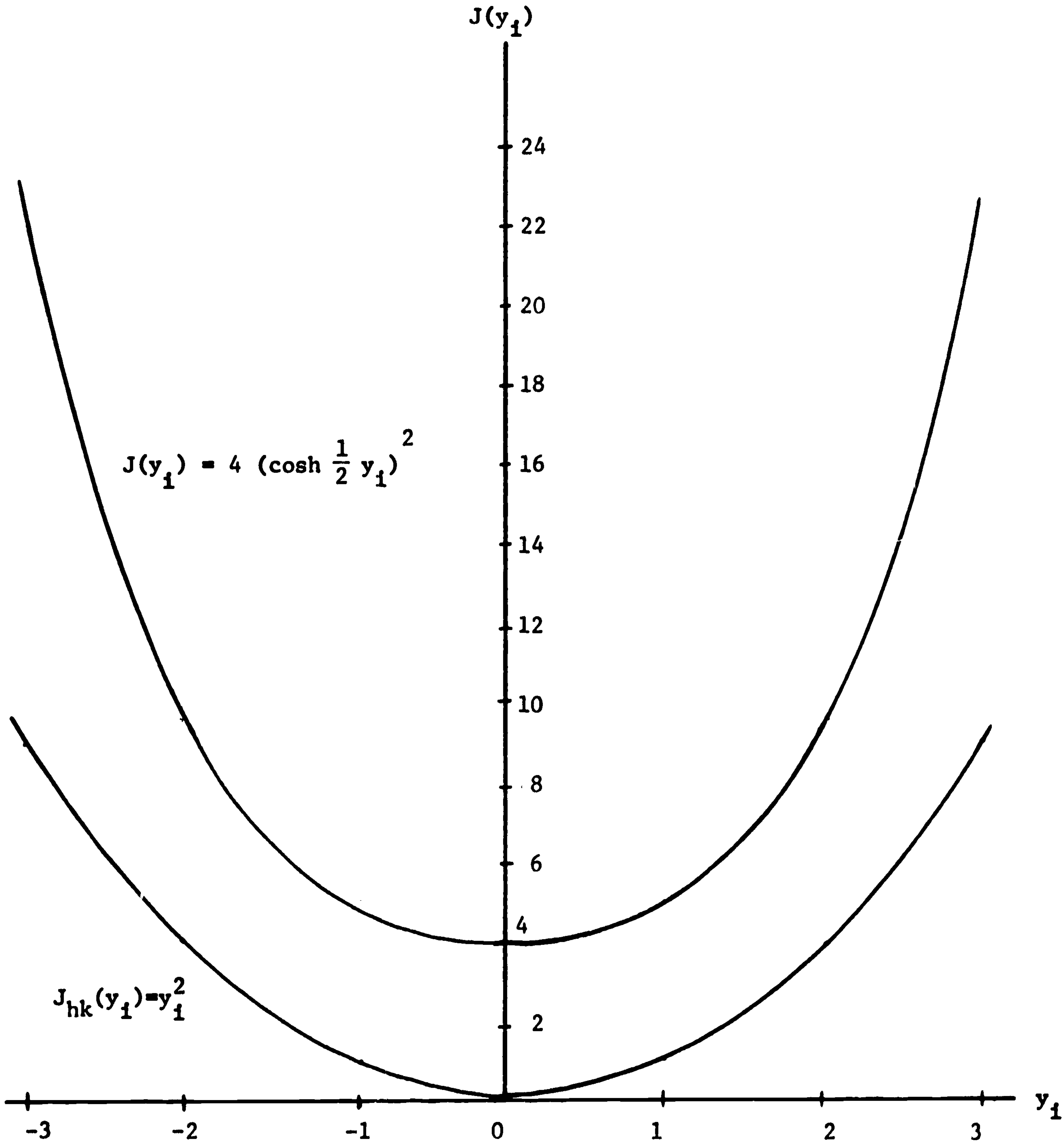


Figure 1. Comparison of Criterion Functions $J(y_i)$ and $J_{hk}(y_i)$.

general, the gradient $\frac{\partial J(\underline{y})}{\partial \underline{y}}$ is greater than the gradient $\frac{\partial J_{hk}(\underline{y})}{\partial \underline{y}}$ everywhere except at the origin $\underline{y} = \underline{0}$. Since the gradient descent procedure is used in both algorithms, and since $\underline{y}$ and $\underline{b}$, or $\underline{y}$ and $\underline{w}$, are linearly related, it is conceivable that the proposed algorithm may have a higher convergence rate for a solution $\underline{w}$.

As mentioned before, $J(\underline{y})$ reaches a minimum when each term $(\cosh \frac{1}{2} y_i)^2$, $(i=1,\ldots,N)$, is minimized. For each $(\cosh \frac{1}{2} y_i)^2$ to be a minimum, each y_i, $(i=1,\ldots,N)$, must equal zero and $\underline{y} = \underline{0}$ gives a desired solution. Thus one is attempting to cluster the values $[_i\underline{x}_1]^t \underline{w}$ and $-[_j\underline{x}_2]^t \underline{w}$, $(i=1,2,\ldots,n_1;j=1,2,\ldots,n_2)$ about the positive scalers b_i's, $(i=1,2,\ldots,N)$. Since the b_i's are only constrained to be positive, $J(\underline{y})$ can be minimized with respect to both $\underline{w}$ and b subject to the condition that $\underline{b} > \underline{0}$. Note that it is not necessary to attain the minimum value of $J(\underline{y})$; in fact, a solution $\underline{w}^*$ is obtained whenever $\underline{y} \geq \underline{0}$ with $\underline{b} > \underline{0}$ from which follows $\underline{A}\,\underline{w}^* \geq \underline{b} > \underline{0}$.

Let the matrix $\underline{A}$ defined in (1.5) be also represented as

$$\underline{A} = \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \cdots & \cdots & \cdots & \cdots \\ a_{N1} & a_{N2} & \cdots & a_{Nn} \end{bmatrix}. \tag{2.9}$$

From (2.3),

$$y_i = a_{i1} w_1 + a_{i2} w_2 + \ldots + a_{in} w_n - b_i \tag{2.10}$$

$$(i=1,2,\ldots,N)$$

and

$$\frac{\partial y_i}{\partial w_j} = a_{ij} \tag{2.11}$$

$$\frac{\partial y_i}{\partial b_j} = -\delta_{ij} \tag{2.12}$$

where δ_{ij} is the kronecker delta. Let

$$J_i(\underline{y}) = 4(\cosh \frac{1}{2} y_i)^2, \quad (i=1,2,\ldots,N) \tag{2.13}$$

then

$$J(\underline{y}) = \sum_{i=1}^{N} J_i(y) \tag{2.14}$$

The gradients of $J_i(\underline{y})$ with respect to w_j and b_j are respectively

$$\frac{\partial J_i(\underline{y})}{\partial w_j} = 4 \ (\cosh \frac{1}{2} y_i)(\sinh \frac{1}{2} y_i) \ \frac{\partial y_i}{\partial w_j}$$

$$= 2 \ (\sinh y_i) \ \frac{\partial y_i}{\partial w_j} = 2 \ a_{ij} \ \sinh y_i \tag{2.15}$$

and

$$\frac{\partial J_i(\underline{y})}{\partial b_j} = 2 \ (\sinh y_i) \ \frac{\partial y_i}{\partial b_j} = -2\delta_{ij} \ \sinh y_i \tag{2.16}$$

Now,

$$\frac{\partial J_i(\underline{y})}{\partial \underline{w}} = \begin{bmatrix} \frac{\partial J_i(\underline{y})}{\partial w_1} \\ \vdots \\ \frac{\partial J_i(\underline{y})}{\partial w_n} \end{bmatrix} = 2 \sinh y_i \begin{bmatrix} a_{i1} \\ \vdots \\ a_{in} \end{bmatrix} \quad (2.17)$$

and

$$\frac{\partial J_i(\underline{y})}{\partial \underline{b}} = \begin{bmatrix} \frac{\partial J_i(\underline{y})}{\partial b_1} \\ \vdots \\ \frac{\partial J_i(\underline{y})}{\partial b_N} \end{bmatrix} = -2 \sinh y_i \begin{bmatrix} \delta_{i1} \\ \vdots \\ \delta_{iN} \end{bmatrix} \quad (2.18)$$

where the derivative of a scalar with respect to a column vector is a column vector. Hence, the gradient of $J(\underline{y})$ with respect to $\underline{w}$ is given by

$$\begin{aligned} \frac{\partial J(\underline{y})}{\partial \underline{w}} &= \sum_{i=1}^{N} \frac{\partial J_i(\underline{y})}{\partial \underline{w}} \\ &= 2 \sinh y_1 \begin{bmatrix} a_{11} \\ a_{12} \\ \vdots \\ a_{1n} \end{bmatrix} + 2 \sinh y_2 \begin{bmatrix} a_{21} \\ a_{22} \\ \vdots \\ a_{2n} \end{bmatrix} + \dots + 2 \sinh y_N \begin{bmatrix} a_{N1} \\ a_{N2} \\ \vdots \\ a_{Nn} \end{bmatrix} \\ &= 2 \begin{bmatrix} a_{11} & a_{21} & \cdots & a_{N1} \\ \dots & \dots & \dots & \dots \\ a_{1n} & a_{2n} & \cdots & a_{Nn} \end{bmatrix} \begin{bmatrix} \sinh y_1 \\ \vdots \\ \sinh y_N \end{bmatrix} \\ &= 2 \underline{A}^t \begin{bmatrix} \sinh y_1 \\ \sinh y_2 \\ \vdots \\ \sinh y_N \end{bmatrix} ; \end{aligned} \quad (2.19)$$

and the gradient of $J(\underline{y})$ with respect to $\underline{b}$ is given by

$$\frac{\partial J(\underline{y})}{\partial \underline{b}} = \sum_{i=1}^{N} \frac{\partial J_i(\underline{y})}{\partial \underline{b}} = -2\underline{I} \begin{bmatrix} \sinh y_1 \\ \sinh y_2 \\ \vdots \\ \sinh y_N \end{bmatrix} = -2 \begin{bmatrix} \sinh y_1 \\ \sinh y_2 \\ \vdots \\ \sinh y_N \end{bmatrix} . \qquad (2.20)$$

Since $\underline{w}$ is not constrained in any way, $\frac{\partial J(\underline{y})}{\partial \underline{w}} = 0$ implies

$$\begin{bmatrix} \sinh y_1 \\ \sinh y_2 \\ \vdots \\ \sinh y_N \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 0 \end{bmatrix}$$

which, in turn, implies $y_i = 0$ for all $i=1,2,\ldots,N$. Therefore, for a fixed $\underline{b} > \underline{0}$, minimizing $J(\underline{y})$ with respect to $\underline{w}$ gives

$$\underline{y} = \underline{A}\,\underline{w} - \underline{b} = \underline{0}.$$

Solving the above equation for $\underline{w}$, one obtains

$$\underline{w} = \underline{A}^{\#}\underline{b} \qquad (2.21)$$

where $\underline{A}^{\#}$ is the generalized inverse of $\underline{A}$[20].

On the other hand, for a fixed $\underline{w}$, $\frac{\partial J(\underline{y})}{\partial \underline{b}} = 0$ with $\underline{b} > \underline{0}$ dictates a descent procedure of the following form, with k denoting the iteration number:

$$\underline{b}(k+1) = \underline{b}(k) + \Delta\underline{b}(k) \tag{2.22}$$

where the components of $\Delta b_i(k)$, $i=1,2,\ldots,N$, of $\Delta\underline{b}(k)$ are governed by

$$\Delta b_i(k) \propto \begin{cases} -\left(\dfrac{\partial J(y(k))}{\partial \underline{b}}\right)_i = 2 \sinh y_i & \text{if } y_i > 0, \\ 0 & \text{if } y_i \leq 0 . \end{cases} \tag{2.23}$$

Introduce a positive scalar $p(k)$ as the proportionality constant and rewrite equation (2.23) in the vector form,

$$\Delta\underline{b}(k) = p(k) \begin{bmatrix} \sinh y_1(k) + |\sinh y_1(k)| \\ \sinh y_2(k) + |\sinh y_2(k)| \\ \vdots \\ \sinh y_N(k) + |\sinh y_N(k)| \end{bmatrix}$$

$$= p(k)\, \underline{h}(k) , \tag{2.24}$$

where

$$\underline{h}(k) = \begin{bmatrix} h_1(k) \\ h_2(k) \\ \vdots \\ h_N(k) \end{bmatrix} = \begin{bmatrix} \sinh y_1(k) + |\sinh y_1(k)| \\ \sinh y_2(k) + |\sinh y_2(k)| \\ \vdots \\ \sinh y_N(k) + |\sinh y_N(k)| \end{bmatrix} . \tag{2.25}$$

As can be shown later, $p(k)$ may be chosen as equal to

$$p(k) = \frac{1}{\cosh y_{max}(k)} \tag{2.26}$$

where

$$y_{max}(k) = \max_i |y_i(k)| . \tag{2.27}$$

Substituting (2.24) into (2.22) and, from (2.21), writing

$$\begin{aligned} \underline{w}(k+1) &= \underline{A}^{\#}\underline{b}(k+1) = \underline{A}^{\#}[\underline{b}(k) + \Delta\underline{b}(k)] \\ &= \underline{w}(k) + p(k)\,\underline{A}^{\#}\,\underline{h}(k) , \end{aligned} \tag{2.28}$$

one obtains the following algorithm:

$$\begin{cases} \underline{w}(0) = \underline{A}^{\#}\underline{b}(0), \qquad \underline{b}(0) > 0 \text{ but otherwise arbitrary} \\ \underline{y}(k) = \underline{A}\,\underline{w}(k) - \underline{b}(k) \\ \underline{b}(k+1) = b(k) + p(k)\,\underline{h}(k) \\ \underline{w}(k+1) = \underline{w}(k) + p(k)\underline{A}^{\#}\underline{h}(k) \end{cases} \tag{2.29}$$

where $\underline{h}(k)$ and $p(k)$ are given by equations (2.25) and (2.26) respectively. Note that in this algorithm $p(k)$ varies at each step and is a nonlinear function of $\underline{y}(k)$. A recursive relation in $y(k)$ can also be obtained from (2.29).

$$\begin{aligned} \underline{y}(k+1) &= \underline{A}\,\underline{w}(k+1) - \underline{b}(k+1) = \underline{A}\,\underline{A}^{\#}\underline{b}(k+1) - \underline{b}(k+1) \\ &= \underline{A}\,\underline{A}^{\#}[\underline{b}(k) + p(k)\underline{h}(k)] - \underline{b}(k) - p(k)\underline{h}(k) \\ &= \underline{A}\,\underline{w}(k) - \underline{b}(k) + (\underline{A}\,\underline{A}^{\#} - \underline{I})p(k)\underline{h}(k) \end{aligned}$$

$$\underline{y}(k+1) = \underline{y}(k) + p(k)\,(\underline{A}\,\underline{A}^{\#} - \underline{I})\,\underline{h}(k) . \tag{2.30}$$

Just like the Ho-Kashyap algorithm, it can be shown that the above algorithm (2.29) converges to a solution $\underline{w}^*$ of the system of linear inequalities in a finite number of steps provided that a solution exists, and simultaneously acts as a test for the inconsistency of the linear inequalities. These properties are formally stated in a theorem as given in the next section.

B. Theorem 1

Before discussing the main theorem, a lemma to be used in the proof of the theorem will be given first.

Lemma 1: Let one consider the set of linear inequalities (2.1) and the algorithm (2.29) to solve this set. Then

1) $\underline{y}(k) \not\leq \underline{0}$ for any k;

and

2) if the set of linear inequalities is consistent, then

$$\underline{y}(k) \not\leq \underline{0} \text{ for any } k.$$

This lemma is the same as the one given by Ho and Kashyap[12] except that the iterative algorithm is different. The proof of the lemma is not given here since it is identical to the proof of Ho-Kashyap lemma. Recall again the notation used in the lemma: $\underline{y}(k) \leq 0$ means that $y_i(k) \leq 0$ for all i but $\underline{y}$ possesses at least one negative component. This lemma is a rigorous statement that with a consistent set of linear inequalities $\underline{A}\ \underline{w} > 0$, the elements of the vector $\underline{y}(k)$ cannot be all non-positive.

<u>Theorem 1</u>: Consider the set of linear inequalities (2.1) and the algorithm (2.29) to solve these inequalities, and let $V[\underline{y}(k)] = ||\underline{y}(k)||^2$.

1) If the set of linear inequalities is consistent then

   a) $\Delta V[\underline{y}(k)] \triangleq V[\underline{y}(k+1)] - V[\underline{y}(k)] < 0$ and $\lim_{k\to\infty} V[\underline{y}(k)] = 0$ implying convergence to a solution in an infinite number of steps; and

   b) actually, a solution is obtained in a finite number of steps.

2) If the set of linear inequalities is inconsistent, then there exist a positive integer k^* such that

$$\Delta V[\underline{y}(k)] < 0 \text{ for } k < k^*$$
$$\Delta V[\underline{y}(k)] = 0 \text{ for } k \geq k^*, \text{ and}$$

$$\underline{y}(k) \nleq \underline{0} \text{ for } k < k^*$$
$$\underline{y}(k) = \underline{y}(k^*) \leq \underline{0} \text{ for } k \geq k^*$$

and

$$\underline{w}(k) = \underline{w}(k^*) \text{ for } k \geq k^*$$
$$\underline{b}(k) = \underline{b}(k^*) \text{ for } k \geq k^*.$$

In other words, the occurrence of a nonpositive vector $\underline{y}(k)$ at any step terminates the algorithm and indicates the inconsistency of the given set of linear inequalities.

Proof:

Part 1: Since the algorithm (2.29) can be rewritten as a recursive relation in $\underline{y}(k)$ given by (2.30), and

$$V[\underline{y}(k)] = ||\underline{y}(k)||^2 > 0 \text{ for all } \underline{y}(k) \neq \underline{0}. \qquad (2.31)$$

$V[\underline{y}(k)]$ can be considered as a Liapunov function for the nonlinear difference equation (2.30). Thus

$$\begin{aligned}
\Delta V[\underline{y}(k)] &\triangleq V[\underline{y}(k+1)] - V[\underline{y}(k)] \\
&= ||\underline{y}(k+1)||^2 - ||\underline{y}(k)||^2 = \underline{y}^t(k+1)\ \underline{y}(k+1) - \underline{y}^t(k)\ \underline{y}(k) \\
&= [\underline{y}(k) + p(k)(\underline{A}\ \underline{A}^{\#} - \underline{I})\underline{h}(k)]^t[\underline{y}(k) + p(k)(\underline{A}\ \underline{A}^{\#} - \underline{I})\underline{h}(k)] \\
&\quad - \underline{y}^t(k)\ \underline{y}(k) \\
&= p(k)\underline{h}^t(k)\ (\underline{A}\ \underline{A}^{\#} - \underline{I})^t\ \underline{y}(k) + p(k)\ y^t(k)\ (\underline{A}\ \underline{A}^{\#} - \underline{I})\ \underline{h}(k) \\
&\quad + p^2(k)\ \underline{h}^t(k)\ (\underline{A}\ \underline{A}^{\#} - \underline{I})^t\ (\underline{A}\ \underline{A}^{\#} - \underline{I})\ \underline{h}\ (k).
\end{aligned}$$

Since $(\underline{A}\ \underline{A}^{\#} - \underline{I})$ is hermitian idempotent[20],

$$(\underline{A}\ \underline{A}^{\#} - \underline{I})^t = (\underline{A}\ \underline{A}^{\#} - \underline{I}),$$

$$\begin{aligned}
(\underline{A}\ \underline{A}^{\#} - \underline{I})^t\ (\underline{A}\ \underline{A}^{\#} - \underline{I}) &= (\underline{A}\ \underline{A}^{\#} - \underline{I})(\underline{A}\ \underline{A}^{\#} - \underline{I}) = \underline{A}\ \underline{A}^{\#}\underline{A}\ \underline{A}^{\#} - \underline{A}\ \underline{A}^{\#} - \underline{A}\ \underline{A}^{\#} + \underline{I} \\
&= \underline{A}\ \underline{A}^{\#} - \underline{A}\ \underline{A}^{\#} - \underline{A}\ \underline{A}^{\#} + \underline{I} = \underline{I} - \underline{A}\ \underline{A}^{\#}.
\end{aligned}$$

then,

$$V[\underline{y}(k)] = 2\ p(k)\ \underline{h}^t(k)\ (\underline{A}\ \underline{A}^{\#} - \underline{I})\underline{y}(k) + p^2(k)\underline{h}^t(k)(\underline{I} - \underline{A}\ \underline{A}^{\#})\underline{h}(k).$$

Now

$$\underline{A}\,\underline{A}^{\#}\underline{y}(k) = \underline{A}\,\underline{A}^{\#}[\underline{A}\,\underline{w}(k) - \underline{b}(k)] = \underline{A}\,\underline{A}^{\#}[\underline{A}\,\underline{A}^{\#}\underline{b}(k) - \underline{b}(k)]$$

$$= [\underline{A}\,\underline{A}^{\#}\underline{A}\,\underline{A}^{\#} - A\,A^{\#}]\underline{b}(k) = [\underline{A}\,\underline{A}^{\#} - \underline{A}\,\underline{A}^{\#}]\underline{b}(k)$$

$$= \underline{0},$$

hence

$\Delta V[\underline{y}(k)]$ reduces to

$$\Delta V[\underline{y}(k)] = -2\,p(k)\underline{h}^t(k)\;\underline{y}(k) + p^2(k)\underline{h}^t(k)\;(\underline{I} - \underline{A}\,\underline{A}^{\#})\underline{h}(k). \qquad (2.32)$$

Let

$$\underline{s}(\underline{y}) \triangleq \begin{bmatrix} \sinh y_1 \\ \sinh y_2 \\ \vdots \\ \sinh y_N \end{bmatrix} = \begin{bmatrix} \frac{\sinh y_1}{y_1} \cdot y_2 \\ \frac{\sinh y_2}{y_2} \cdot y_2 \\ \vdots \\ \frac{\sinh y_N}{y_N} \cdot y_N \end{bmatrix}$$

$$= \begin{bmatrix} \frac{\sinh y_1}{y_1} & 0 & \cdots & 0 \\ 0 & \frac{\sinh y_2}{y_2} & \cdots & 0 \\ \cdots & \cdots & \cdots & \cdots \\ 0 & 0 & & \frac{\sinh y_N}{y_N} \end{bmatrix} \cdot \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_N \end{bmatrix}$$

$$= \underline{R}(\underline{y})\;\underline{y} \qquad (2.33)$$

where

$$\underline{R}(\underline{y}) \overset{\Delta}{=} \begin{bmatrix} \frac{\sinh y_1}{y_1} & 0 & \cdots & 0 \\ 0 & \frac{\sinh y_2}{y_2} & \cdots & 0 \\ \cdots & \cdots & \cdots & \cdots \\ 0 & 0 & \cdots & \frac{\sinh y_N}{y_N} \end{bmatrix}$$

$$= \operatorname{diag}\,(r_{11}, r_{22}, \ldots, r_{NN}). \tag{2.34}$$

Note that $r_{ii} = \dfrac{\sinh y_i}{y_i} > 0$ for all y_i, $\underline{R}(\underline{y}) > 0$ and $\underline{R}^t(\underline{y}) = \underline{R}(\underline{y})$. Then $\underline{s}(\underline{y})$ has the following properties

$$\underline{s}^t(\underline{y}) = \underline{y}^t\underline{R}^t(\underline{y}) = \underline{y}^t\underline{R}(\underline{y}),$$

$$|\underline{s}(\underline{y})| = |\underline{R}(\underline{y})\underline{y}| = \underline{R}(\underline{y})\,|\underline{y}|,$$

$$|\underline{s}(\underline{y})|^t = |\underline{s}^t(\underline{y})| = |\underline{y}|^t\,\underline{R}(\underline{y})\ . \tag{2.35}$$

From (2.25) and (2.35), the properties of $\underline{h}(\underline{y})$ are:

$$\underline{h}(\underline{y}) = \underline{s}(\underline{y}) + |\underline{s}(\underline{y})| = \underline{R}(\underline{y})\underline{y} + \underline{R}(\underline{y})\,|\underline{y}| = \underline{R}(\underline{y})[\underline{y} + |\underline{y}|],$$

$$\underline{h}^t(\underline{y}) = [\underline{s}(\underline{y}) + |\underline{s}(\underline{y})|]^t = \underline{s}^t(\underline{y}) + |\underline{s}(\underline{y})|^t = \underline{y}^t\underline{R}(\underline{y}) + |\underline{y}|^t\underline{R}(\underline{y})$$
$$= [\underline{y} + |\underline{y}|]^t\,\underline{R}(\underline{y}). \tag{2.36}$$

Reducing the first term of equation (2.32) by the relation in (2.36), one obtains

$$
\begin{aligned}
-2\ p(k)\underline{h}^t(k)\underline{y}(k) &= -2\ p(k)[\underline{y}(k) + |y(k)|]^t\ \underline{R}(k)\ \underline{y}(k) \\
&= -p(k)[\underline{y}(k) + |\underline{y}(k)|]^t\ \underline{R}(k)\ \underline{y}(k) \\
&\quad -p(k)[\underline{y}(k) + |y(k)|]^t\ \underline{R}(k)\underline{y}(k).
\end{aligned}
$$

Adding $|\underline{y}(k)|$ to the first term and subtracting $|\underline{y}(k)|$ from the second term on the right hand side of the above equation gives

$$
\begin{aligned}
-2\ p(k)\underline{h}^t(k)\ \underline{y}(k) &= -p(k)[\underline{y}(k) + |\underline{y}(k)|]^t\ \underline{R}(k)\ [\underline{y}(k) + |\underline{y}(k)|] \\
&\quad -p(k)[\underline{y}(k) + |\underline{y}(k)|]^t\ \underline{R}(k)[\underline{y}(k) - |\underline{y}(k)|]. \qquad (2.37)
\end{aligned}
$$

It will be shown that the second term on the right hand side of equation (2.37) is zero. Since

$$
\begin{aligned}
&[\underline{y}(k) + |\underline{y}(k)|]^t\ \underline{R}(k)\ [\underline{y}(k) - \underline{y}(k)] = \\
&\quad = [y_1(k) + |y_1(k)|, \ldots, y_N(k) + |y_N(k)|]\cdot \\
&\quad \cdot \begin{bmatrix} r_{11}(k) & 0 & \cdots & 0 \\ 0 & r_{22}(k) & \cdots & 0 \\ \vdots & \vdots & & \vdots \\ 0 & 0 & \cdots & r_{NN}(k) \end{bmatrix} \cdot \begin{bmatrix} y_1(k) - |y_1(k)| \\ y_2(k) - |y_2(k)| \\ \cdots\cdots \\ y_N(k) - |y_N(k)| \end{bmatrix} \\
&\quad = \sum_{i=1}^{N} r_{ii}(k)[y_i(k) + |y_i(k)|][y_i(k) - |y_i(k)|] \qquad (2.38)
\end{aligned}
$$

and

$$[y_i(k) - |y_i(k)|] = 0. \text{ if } y_i(k) \geq 0$$

$$[y_i(k) + |y_i(k)|] = 0. \text{ if } y_i(k) \leq 0,$$

therefore,

$$[\underline{y}(k) + |\underline{y}(k)|]^t \underline{R}(k)[\underline{y}(k) - \underline{y}(k)] = 0 . \qquad (2.39)$$

Substitute (2.39) into (2.37). The first term of equation (2.32) is then reduced to

$$-2\, p(k)\underline{h}^t(k)\ \underline{y}(k) = -p(k)[\underline{y}(k) + |\underline{y}(k)|]^t \underline{R}(k)[\underline{y}(k) + |\underline{y}(k)]. (2.40)$$

Substituting (2.36) and (2.40) into (2.32), one obtains

$$\begin{aligned}\Delta V[\underline{y}(k)] &= -p(k)[\underline{y}(k) + |\underline{y}(k)|]^t \underline{R}(k)[\underline{y}(k) + |\underline{y}(k)|] \\ &\quad + p^2(k)[\underline{y}(k) + |\underline{y}(k)|]^t \underline{R}(k)\ (\underline{I} - \underline{A}\ \underline{A}^{\#})\underline{R}(k)[\underline{y}(k) + |y(k)|] \\ &= -[\underline{y}(k) + |\underline{y}(k)|]^t [p(k)\underline{R}(k) + p^2(k)\underline{R}(k)(\underline{A}\ \underline{A}^{\#} - \underline{I})\underline{R}(k)][\underline{y}(k) + |\ \underline{y}(k)|] \\ &= -||\underline{y}(k) + |\underline{y}(k)|||^2 \\ &\qquad \{p^2(k)\underline{R}(k)\underline{A}\ \underline{A}^{\#}\underline{R}(k) + p(k)\underline{R}(k) - p^2(k)\underline{R}^2(k)\}.\end{aligned} \qquad (2.41)$$

For $\Delta V[\underline{y}(k)]$ to be negative semidefinite, in particular, $\Delta V\{\underline{y}(k)] = 0$ only if $\underline{y}(k) = 0$ or $\underline{y}(k) \leq 0$, the matrix $[p^2(k)\underline{R}(k)\ \underline{A}\ \underline{A}^{\#}\underline{R}(k) + p(k)\underline{R}(k) - p^2(k)\underline{R}^2(k)]$

must be positive definite. $\underline{A}\,\underline{A}^{\#}$ is positive semidefinite since $\underline{A}\,\underline{A}^{\#}$ is hermitian idempotent, $\underline{x}^t\underline{A}\,\underline{A}^{\#}\,\underline{x} \geq 0$ for any x; it follows that $\underline{x}^t\underline{R}\,\underline{A}\,\underline{A}^{\#}\underline{R}\,\underline{x} = \underline{x}^t\underline{R}^t\underline{A}\,\underline{A}^{\#}\underline{R}\,\underline{x} = (\underline{R}\,\underline{x})^t\underline{A}\,\underline{A}^{\#}(\underline{R}\,\underline{x}) = \underline{z}^t\underline{A}\,\underline{A}^{\#}\underline{z} \geq 0$ for any $\underline{z}$; hence $\underline{R}\,\underline{A}\,\underline{A}^{\#}\underline{R}$ is also positive semidefinite. Now one can choose a p(k) such that $[p(k)\underline{R}(k) - p^2(k)\underline{R}^2(k)]$ is positive definite. From (2.34),

$$p(k)\underline{R}(k)-p^2(k)\underline{R}^2(k) =$$

$$= \begin{matrix} p(k)r_{11}(k) - p^2(k)r_{11}^2(k) & 0 & \cdots & 0 \\ 0 & p(k)r_{22}(k)-p^2(k)r_{22}^2(k) & & 0 \\ \vdots & \vdots & & \vdots \\ 0 & 0 & & p(k)r_{NN}(k)-p^2(k)r_{NN}^2(k) \end{matrix}$$

$[p(k)\underline{R}(k) - p^2(k)\underline{R}^2(k)]$ is positive definite if

$$[p(k)r_{ii}(k) - p^2(k)r_{ii}^2(k)] > 0 \text{ for all } i=1,2,\ldots,N. \tag{2.42}$$

Since $r_{ii}(k) = \dfrac{\sinh y_i}{y_i} > 0$ for all i and p(k) is restricted to be positive, the above condition reduces to the condition

$$1 - p(k)\, r_{ii}(k) > 0 \text{ for all } i=1,2,\ldots,N. \tag{2.43}$$

For p(k) chosen in equation (2.26),

$$p(k) = \frac{1}{\cosh y_{max}(k)}$$

where

$$y_{max}(k) = \operatorname*{Max}_{i} |y_i(k)|,$$

$$p(k)r_{ii}(k) = \frac{1}{\cosh y_{max}(k)} \cdot \frac{\sinh y_i(k)}{y_i(k)} = \frac{\sinh y_i(k)}{y_i(k)\cosh y_{max}(k)}$$

$$= \frac{(y_i(k) + \frac{y_i^3(k)}{3!} + \frac{y_i^5(k)}{5!} + \ldots)}{y_i(k)(1 + \frac{y_{max}^2(k)}{2!} + \frac{y_{max}^4(k)}{4!} + \ldots)} = \frac{(1 + \frac{y_i^2(k)}{3!} + \frac{y_i^4(k)}{5!} + \ldots)}{(1 + \frac{y_{max}^2}{2!} + \frac{y_{max}^4}{4!} + \ldots)}$$

$$= \frac{\sum_{n=o}^{\infty} \frac{y_i^{2n}(k)}{(2n+1)!}}{\sum_{n=o}^{\infty} \frac{y_{max}^{2n}(k)}{(2n)!}}$$

Note that

$$\frac{y_i^2(k)}{y^2_{max}(k)} \leq 1 \text{ for all } i=1,\ldots N$$

$$\frac{\frac{y_i^{2n}(k)}{(2n+1)!}}{\frac{y_{max}^{2n}(k)}{(2n)!}} < 1 \text{ for all } \begin{matrix} i=1,2,\ldots,N \\ n=1,2,\ldots,\infty \end{matrix}$$

it follows that

$$p(k)r_{ii}(k) = \frac{\sum_{n=o}^{\infty} \frac{y_i^{2n}(k)}{(2n+1)!}}{\sum_{n=o}^{\infty} \frac{y_{max}^{2n}(k)}{(2n)!}} < 1 . \tag{2.44}$$

Thus the condition (2.43) is satisfied and $[p(k)\underline{R}(k) - p^2(k)\underline{R}^2(k)]$ is positive definite for $p(k) = \frac{1}{\cosh y_{max}(k)}$. Thus $\Delta V[\underline{y}(k)]$ has the desired property of negative semidefinite for $p(k) = \frac{1}{\cosh y_{max}(k)}$ and for any finite $\underline{y}(k)$.

From equation (2.41) one notes that $\Delta V[\underline{y}(k)]$ equals zero if and only if $\underline{y}(k) = \underline{0}$ or $\underline{y}(k) \leq \underline{0}$. Since it is assumed that the set of linear inequalities (2.1) is consistent, and from the lemma $\underline{y}(k) \nleq \underline{0}$, therefore

$$\begin{aligned} \Delta V[\underline{y}(k)] &< 0 \quad \text{for all } \underline{y}(k) \neq \underline{0} \\ &= 0 \quad \text{if } \underline{y}(k) = \underline{0} . \end{aligned} \tag{2.45}$$

By Liapunov's stability criterion, the equilibrium state $\underline{y} = \underline{0}$ of the discrete system (2.30) can be reached asymptotically, i.e., $\lim_{k\to\infty} ||\underline{y}(k)||^2 = 0$, which corresponds to a solution $\underline{w}^{**}$ with $\underline{A}\,\underline{w}^{**} = \underline{b} > \underline{0}$. This completes the proof of Part 1(a).

To prove the convergence of the algorithm (2.29) in a finite number of steps, one notes that $\underline{b}(k)$ is a nondecreasing vector. Let

$$\underline{b}^t(0) = [1,1,\ldots 1]$$

then

$$\underline{b}^t(k) \geq \underline{b}^t(0) \geq [1,1,\ldots,1] \text{ for any } k > 0.$$

Since $\underline{A}\,\underline{w}(k) = \underline{b}(k) + \underline{y}(k)$, $|\underline{y}^t(k)| < [1,1,\ldots,1]$ implies $\underline{A}\,\underline{w}^*(k) > \underline{0}$ when a solution w^* is reached. But $V[\underline{y}(k)] \leq 1$ implies $|\underline{y}^t(k)| < [1,\ldots,1]$. Since $V[\underline{y}(k)]$ converges to zero in infinite time, it must converge to the region $V[\underline{y}(k)] = 1$ in finite time, hence $|\underline{y}^t(k)| < [1,1,\ldots,1]$, $\underline{A}\,\underline{w}(k) > 0$, and a solution $\underline{w}^* = \underline{w}(k)$ is obtained in a finite number of steps. This completes the proof of Part 1(b).

Part 2: It has been proved in Part 1 that $V[\underline{y}(k)]$ is negative semidefinite independent of the consistency of the linear inequalities. Now, if the set of linear inequalities (2.1) is inconsistent, one notes that $\underline{y}(k)$ cannot be $\underline{0}$ and hence $V[\underline{y}(k)]$ cannot become zero for any $k > 0$. There must exist a value of k, called k^*, such that

$$\Delta V[\underline{y}(k)] < 0 \quad \text{for } 0 \leq k < k^*,$$

$$= 0 \quad \text{for } k = k^*,$$

$$\underline{y}(k) \not\leq \underline{0} \quad \text{for } 0 \leq k < k^*.$$

But $V[\underline{y}(k^*)] = 0$ if either $\underline{y}(k^*) = \underline{0}$ or $\underline{y}(k^*) \leq \underline{0}$. Since $\underline{y}(k^*) \neq \underline{0}$, this implies $\underline{y}(k^*) \leq 0$ and hence, from (2.25), $\underline{h}(k^*) = \underline{0}$. Equation (2.30) indicates that

$$\underline{y}(k) = \underline{y}(k^*) \leq \underline{0} \text{ for all } k \geq k^*$$

As a consequence, one obtains

$$\Delta V[\underline{y}(k)] = 0 \qquad \text{for all } k \geq k^*$$

$$\underline{h}(k) = 0 \qquad \text{for all } k \geq k^*$$

$$\underline{w}(k) = \underline{w}(k^*) \qquad \text{for all } k \geq k^*$$

$$\underline{b}(k) = \underline{b}(k^*) \qquad \text{for all } k \geq k^*$$

This completes the proof of the theorem.

C. An Optimum Choice of p(k)

The choice of $p(k) = \frac{1}{\cosh y_{max}(k)}$ in the previous section is only one of many possible choices of p(k) for the convergence of the algorithm (2.29). The convergence rate may be further improved by choosing a p(k) such that the decrease in the Liapunov function $V[\underline{y}(k)]$ is maximized at every step, that is, $-\Delta V[\underline{y}(k)]$ is maximized with respect to p(k).

Take the partial derivative of $\Delta V[\underline{y}(k)]$ in equation (2.41) with respect to p(k),

$$\frac{\partial\{-\Delta V[\underline{y}(k)]\}}{\partial\{p(k)\}} = \frac{\partial\{[\underline{y}(k)+|\underline{y}(k)|]^t[p(k)\underline{R}(k)+p^2(k)\underline{R}(k)(\underline{AA}^{\#}-\underline{I})\underline{R}(k)][\underline{y}(k)+|\underline{y}(k)|]}{\partial p(k)}$$

$$= [\underline{y}(k)+|\underline{y}(k)|]^t[R(k)-2p(k)\underline{R}(k)(\underline{I}-\underline{AA}^{\#})\underline{R}(k)][\underline{y}(k)+|\underline{y}(k)|]. \qquad (2.46)$$

For $-\Delta V[\underline{y}(k)]$ to be a maximum, $\frac{\partial\{-\Delta V[\underline{y}(k)]\}}{\partial p(k)}$ must equal zero as a necessary condition. Hence,

$$2p(k)[\underline{y}(k) + |\underline{y}(k)|]^t[\underline{R}(k)(\underline{I} - \underline{A}\,\underline{A}^{\#})\underline{R}(k)][\underline{y}(k) + |\underline{y}(k)|]$$

$$= [\underline{y}(k) + |\underline{y}(k)|]^t\underline{R}(k)[\underline{y}(k) + |\underline{y}(k)|]$$

$$p(k) = \frac{[\underline{y}(k) + |\underline{y}(k)|]^t\underline{R}(k)[\underline{y}(k) + |\underline{y}(k)|]}{2[\underline{y}(k) + |\underline{y}(k)|]^t\underline{R}(k)[\underline{I} - \underline{A}\,\underline{A}^{\#}]\underline{R}(k)[\underline{y}(k) + |\underline{y}(k)|]} \tag{2.47}$$

provided that

$$[\underline{y}(k) + |\underline{y}(k)|]^t\underline{R}(k)[\underline{I} - \underline{A}\,\underline{A}^{\#}]\underline{R}(k)[\underline{y}(k) + |\underline{y}(k)|] \neq 0. \tag{2.48}$$

During the iteration process, $\underline{y}(k) \neq \underline{0}$ and $\underline{y}(k) \nleq \underline{0}$. Since $\underline{R}(k) > 0$ and $\underline{I} - \underline{A}\,\underline{A}^{\#} \geq 0$, the condition (2.48) is satisfied unless $\underline{I} - \underline{A}\,\underline{A}^{\#} = \underline{0}$; therefore, for $\underline{I} - \underline{A}\,\underline{A}^{\#} > 0$, both numerator and denominator in (2.47) are positive definite, hence p(k) given by (2.47) is positive. At this value of p(k), $\Delta V[y(k)]$ is negative definite in $[\underline{y}(k) + |\underline{y}(k)|]$ which is required in the convergence proof of the algorithm (2.29). This can be shown by substituting (2.47) into (2.41) which, upon simplification, gives

$$\Delta V[y(k)] = -\frac{1}{2}p(k)[\underline{y}(k) + |\underline{y}(k)|]^t\underline{R}(k)[\underline{y}(k) + |\underline{y}(k)|] < 0 \tag{2.49}$$

For this value of $-\Delta V[\underline{y}(k)]$ to be a maximum, $\dfrac{\partial^2\{-\Delta V[\underline{y}(k)]\}}{\partial\{p(k)\}^2}$ must be less than zero for p(k) given by (2.47). Since, in general,

$$\frac{\partial^2\{-\Delta V[\underline{y}(k)]\}}{\partial\{p(k)\}^2} = [\underline{y}(k) + |\underline{y}(k)|]^t[2\underline{R}(k)(\underline{A}\ \underline{A}^{\#} - \underline{I})\underline{R}(k)[\underline{y}(k) + |\underline{y}(k)|] \tag{2.50}$$

which is negative definite in $[\underline{y}(k) + |\underline{y}(k)|]$. Thus p(k) of equation (2.47) does maximize $-\Delta V[\underline{y}(k)]$ at each iteration and is the optimum choice if $\underline{I} - \underline{A}\ \underline{A}^{\#} > 0$.

If $\underline{I} - \underline{A}\ \underline{A}^{\#} = 0$, equation (2.48) is not satisfied and $-\Delta V[\underline{y}(k)]$ becomes a linear function of p(k),

$$-\Delta V[\underline{y}(k)] = [\underline{y}(k) + |\underline{y}(k)|]^t p(k)\underline{R}(k)[\underline{y}(k) + |\underline{y}(k)|]$$

which has no finite maxima at finite p(k). Equation (2.47) cannot be used but any other positive p(k) greater than $\frac{1}{\cosh y_{max}(k)}$ will improve the convergence rate.

D. Summary of the Procedure

The following ten steps summarize the procedure developed to solve for a solution $\underline{w}$ of a set of linear inequalities $\underline{A}\ \underline{w} > 0$.

1. Select a $\underline{b}(0) > \underline{0}$. Calculate the initial weight vector, $\underline{w}(0)$, where $\underline{w}(0) = \underline{A}^{\#}\underline{b}(0)$.
2. Determine the $\underline{z}$ vector, where $\underline{z} \overset{\Delta}{=} \underline{A}\ \underline{w}$.
3. Check if the $\underline{z}$ vector is greater than $\underline{0}$, that is all $z_i > 0$, for i=1,...,N.

4. If $\underline{z}$ is greater than $\underline{0}$, a solution w has just been obtained and the problem is linear separable; otherwise

5. Calculate the $\underline{y}$ vector by $\underline{y} = \underline{z} - \underline{b}$.

6. Check the $\underline{y}$ vector if $\underline{y} \leq \underline{0}$, that is, all $y_i \leq 0$, for $i=1,\ldots,N$, but with at least one negative component.

7. If $\underline{y} \leq \underline{0}$, then the set of linear inequalities is inconsistent or the problem is not linear separable; otherwise

8. Modify $\underline{b}$ such that $\underline{b} = \underline{b} + p\,\underline{h}$, where $\underline{h}$ is calculated from equation (2.25) and p from either equation (2.26) or equation (2.47).

9. Modify $\underline{w}$ such that $\underline{w} = \underline{w} + p\,\underline{\Lambda}^{\#}\underline{h}$.

10. Return to step 2.

The above steps are shown in the flow chart in Figure 2. Notice that, just like the Ho-Kashyap algorithm, the process continues until the consistency or separability of the problem is determined.

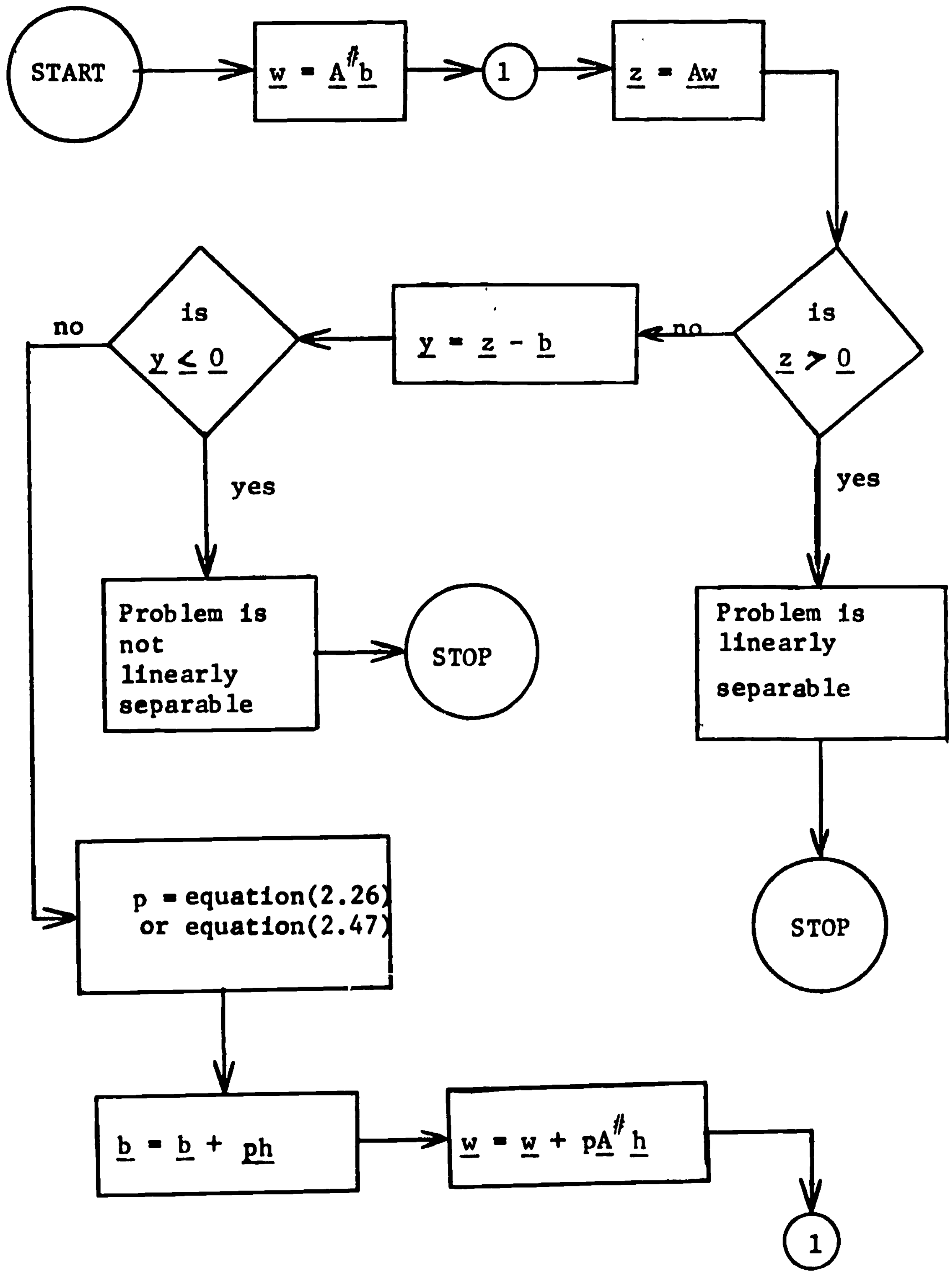


Figure 2. Flow Chart of the Proposed Algorithm.

III. APPLICATION OF THE ACCELERATED ALGORITHM TO SWITCHING FUNCTIONS

A. A Special Algorithm for Switching Functions

For a switching function of r binary variables, $x_1, x_2, \ldots, x_r$, one is concerned with the vertices of a r cube, each vertex being assigned to only one of the two classes C_1 or C_2. It is required to find a separating hyperplane, if one exists, between the two classes.

If an n by 1 vector $\underline{x}$, (n=r+1), as defined in (1.1), is associated with each vertex of the hypercube, that is,

$$\underline{x}^t = (x_0, x_1, x_2, \ldots x_r)$$

where x_0 is the threshold attribute which will always equal +1 and the components $x_1, \ldots, x_r$ are the coordinates of a vertex of the r-dimensional hypercube. Assume that each x_i, (i=1,2,...,r), may take on values +1 and -1 instead of +1 and 0. Let

$$\{{}_1\underline{x}_1,\ {}_2\underline{x}_1,\ \ldots,\ {}_{n_1}\underline{x}_1\} \in \text{Class } C_1$$

$$\{{}_{(n_1+1)}\underline{x}_2,\ {}_{(n_1+2)}\underline{x}_2,\ \ldots,\ {}_m\underline{x}_2\} \in \text{Class } C_2$$

where the intersection of class 1 and class 2 is the empty set. Each one of the 2^r vertices of the r cube is allotted to one class or the other. Then the total number of pattern vectors of the two classes is $m=2^r$.

Finding a separating hyperplane $g(\underline{x}) = 0$ between the two classes is equivalent to finding a weight vector $\underline{w}$ as defined in (1.2) such that

$$g({}_j\underline{x}) = {}_j\underline{x}^t\,\underline{w} \begin{cases} > 0 & \text{for } j=1,2,\ldots,n_1 \\ < 0 & \text{for } j=n_1+1,\ldots,m \end{cases} \tag{3.1}$$

which is the same as

$$\sum_{i=2}^{n} {}_jx_{i-1}w_i \begin{cases} > -w_1 & \text{for } j=1,\ldots,n_1 \\ < -w_1 & \text{for } j=n_1+1,\ldots,m \end{cases}$$

where $-w_1$ is called the threshold value and w_i's, $(i=2,3,\ldots,n)$, are called weights for the switching function $g(\underline{x}) = \underline{x}^t\underline{w}$. Write all ${}_j\underline{x}$ in a compact matrix form as defined in (1.5)

$$\underline{A} = \begin{bmatrix} {}_1\underline{x}^t{}_1 \\ {}_2\underline{x}^t{}_1 \\ \vdots \\ {}_{n_1}\underline{x}^t{}_1 \\ -{}_{(n_1+1)}\underline{x}^t{}_2 \\ -{}_{(n_1+2)}\underline{x}^t{}_2 \\ \vdots \\ -{}_m\underline{x}^t{}_2 \end{bmatrix} \tag{3.2}$$

The set of equations (3.1) can be rewritten as

$$\underline{A}\,\underline{w} > \underline{0}\ .$$

Then a weight vector $\underline{w}$ of the Boolean function $g(\underline{x})$ can be obtained by solving for the above inequalities. If a separating hyperplane or a switching function does not exist, then the above inequalities will be inconsistent.

The accelerated algorithm developed in the previous chapter will be used to obtain a suitable weight vector for each of the switching functions considered in the next section to show its high convergence rate and effectiveness. Following the Ho-Kashyap discussion[12], the algorithm can be significantly simplified, however, owing to the special nature of switching functions. An essential property of the binary variables $x_1 \ldots, x_r$ is normality and orthogonality[12,21], thus

$$\underline{A}^t\underline{A} = 2^r I = 2^{n-1} I \tag{3.3}$$

and

$$\underline{A}^{\#} = (\underline{A}^t\underline{A})^{\#}\underline{A}^t = (2^r\underline{I})^{\#}\,\underline{A}^t = (2^r I)^{-1}\underline{A}^t = 2^{-r}\,\underline{A}^t\ . \tag{3.4}$$

Hence, $\underline{A}^{\#}$ in the algorithm can be replaced by $2^{-r}\,\underline{A}^t$ for switching functions. The accelerated algorithm in equation (2.29) becomes

$$\begin{cases} \underline{w}(0) = 2^{-(n-1)}\underline{A}^t\underline{b}(0), \ \underline{b}(0) > \underline{0} \text{ but otherwise arbitrary} \\ \underline{y}(k) = \underline{A}\ \underline{w}(k) - \underline{b}(k) \\ \underline{b}(k+1) = \underline{b}(k) + p(k)\underline{h}(k) \\ \underline{w}(k+1) = \underline{w}(k) + 2^{-(n-1)}p(k)\underline{A}^t\underline{h}(k) \end{cases} \tag{3.5}$$

where $\underline{h}(k)$ is given by equation (2.25), and $p(k)$ can be given by either equation (2.26) or equation (2.47). A digital computer program for the special algorithm (3.5) has been written in MAD language and is listed in Appendix A.

B. Example Problems

Seven switching function problems are presented to demonstrate the effectiveness of the accelerated algorithm. Comparisons are made between the results obtained by this algorithm and those obtained by Ho-Kashyap algorithm to illustrate the improved convergence rate. The first two examples are explained in detail while the results of the other five examples are given and discussed. Example 3 is a Boolean switching function defined by Winder[6] as a testing function for newly created procedures for switching problems.

1. Example 1: A switching function of three binary variables.

Consider that in a Boolean function of three binary variables A, B, and C,

$$T = A\,B' + A\,C' + B'C'$$

$$F = B\,C + A'C + A'B\ .$$

Designate the true $\underline{x}$'s as of class C_1 and the false $\underline{x}$'s as of class C_2. Then

$$\text{Class } C_1 = \{0,4,5,6\} = \{{}_1\underline{x}_1,\ {}_2\underline{x}_1,\ {}_3\underline{x}_1,\ {}_4\underline{x}_1\}$$

$$\text{Class } C_2 = \{1,2,3,7\} = \{{}_5x_2,\ {}_6x_2,\ {}_7x_2,\ {}_8x_2\}$$

Using (1,-1), instead of (1,0), for the binary representation of x_i, ($i=1,\ldots,r$; $r=3$), one obtains

$${}_1\underline{x}^t_1 = (1,-1,-1,-1)$$

$${}_2\underline{x}^t_1 = (1,1,-1,-1)$$

$${}_3\underline{x}^t_1 = (1,1,-1,1)$$

$${}_4\underline{x}^t_1 = (1,1,1,-1)$$

$${}_5\underline{x}^t_2 = (1,-1,-1,1)$$

$${}_6\underline{x}^t_2 = (1,-1,1,-1)$$

$${}_7\underline{x}^t_2 = (1,-1,1,1)$$

$${}_8\underline{x}^t_2 = (1,1,1,1)$$

Note that x_o always equal to +1 and, in this case, n=4, m=8. The matrix $\underline{A}$ is

$$\underline{A} = \begin{bmatrix} {}_1\underline{x}_1 \\ {}_2\underline{x}_1 \\ {}_3\underline{x}_1 \\ {}_4\underline{x}_1 \\ -{}_5\underline{x}_2 \\ -{}_6\underline{x}_2 \\ -{}_7\underline{x}_2 \\ -{}_8\underline{x}_2 \end{bmatrix} = \begin{bmatrix} 1 & -1 & -1 & -1 \\ 1 & 1 & -1 & -1 \\ 1 & 1 & -1 & 1 \\ 1 & 1 & 1 & -1 \\ -1 & 1 & 1 & -1 \\ -1 & 1 & -1 & 1 \\ -1 & 1 & -1 & -1 \\ -1 & -1 & -1 & -1 \end{bmatrix}.$$

Choose

$$\underline{b}^t(0) = [1,1,1,1,1,1,1,1]$$

then

$$\underline{w}(0) = \frac{1}{2^{n-1}} \underline{A}^t\underline{b}(0) = \frac{1}{8}\begin{bmatrix} 0 \\ 4 \\ -4 \\ -4 \end{bmatrix} = \begin{bmatrix} 0 \\ 1/2 \\ -1/2 \\ -1/2 \end{bmatrix}.$$

and

$$\underline{A}\ \underline{w}(0) = \frac{1}{2}\begin{bmatrix}1\\3\\1\\1\\1\\1\\3\\1\end{bmatrix} > \underline{0}\ .$$

Since $\underline{A}\ \underline{w}(0) > \underline{0}$, the procedure terminates at the zeroth iteration. This result is the same as obtained from the Ho-Kashyap algorithm. A switching function $g(\underline{x}) = \underline{x}^t\underline{w}$ is obtained by taking the threshold element $-w_1 = 0$ and the three weight components $w_2 = \frac{1}{2}$, $w_3 = -\frac{1}{2}$, $w_4 = -\frac{1}{2}$. Note that this $\underline{w}(0)$ should also satisfy the relationships fof the Boolean function T and F and it does as shown below:

$$A\ B' \rightarrow 1/2 - (-1/2) = 1 \ > 0$$

$$A\ C' \rightarrow 1/2 - (-1/2) = 1 \ > 0$$

$$B'C' \rightarrow -(-1/2)-(-1/2) = 1 > 0$$

$$B\ C \rightarrow (-1/2) + (-1/2) = -1 < 0$$

$$A'C \rightarrow -(1/2) + (-1/2) = -1 < 0$$

$$A'B \rightarrow -(1/2) + (-1/2) = -1 < 0$$

2. Example 2: A switching function of four binary variables.

In a Boolean function of four variables A, B, C, and D, consider

$$T = B\ C\ D + A\ C + A\ D + A\ B'$$

$$F = B\ C'D' + A'C' + A'D' + A'B'.$$

This corresponds to Class C_1 and Class C_2 of $\underline{x}$,

$$\text{Class } C_1 = \{7,9 \text{ to } 15\} = \{{}_1\underline{x}_1, \ldots {}_8\underline{x}_1\}$$

$$\text{Class } C_2 = \{0 \text{ to } 6,8\} = {}_9\underline{x}_2, \ldots {}_{16}\underline{x}_2\} .$$

Using (1,-1) for the binary representation of x_i, $(i=1,\ldots,r;r=4)$, one obtains

$${}_1\underline{x}^t_1 = (1,-1,1,1,1)$$

$${}_2\underline{x}^t_1 = (1,1,-1,-1,1)$$

$${}_3\underline{x}^t_1 = (1,1,-1,1,-1)$$

$${}_4\underline{x}^t_1 = (1,1,-1,1,1)$$

$${}_5\underline{x}^t_1 = (1,1,1,-1,-1)$$

$${}_6\underline{x}^t_1 = (1,1,1,-1,1)$$

$${}_7\underline{x}^t_1 = (1,1,1,1,-1)$$

$${}_8\underline{x}^t_1 = (1,1,1,1,1)$$

$${}_9\underline{x}^t_2 = (1,-1,-1,-1,-1)$$

$${}_{10}\underline{x}^t_2 = (1,-1,-1,-1,1)$$

$${}_{11}\underline{x}^t_2 = (1,-1,-1,1,-1)$$

$${}_{12}\underline{x}^t_2 = (1,-1,-1,1,1)$$

$${}_{13}\underline{x}^{t}{}_{2} = (1,-1,1,-1,-1)$$

$${}_{14}\underline{x}^{t}{}_{2} = (1,-1,1,-1,1)$$

$${}_{15}\underline{x}^{t}{}_{2} = (1,-1,1,1,-1)$$

$${}_{16}\underline{x}^{t}{}_{2} = (1,1,-1,-1,-1)$$

The m by n matrix $\underline{A}$, where $n=5$, $m=16$, is represented by

$$\underline{A} = \begin{bmatrix} 1 & -1 & 1 & 1 & 1 \\ 1 & 1 & -1 & -1 & 1 \\ 1 & 1 & -1 & 1 & -1 \\ 1 & 1 & -1 & 1 & 1 \\ 1 & 1 & 1 & -1 & -1 \\ 1 & 1 & 1 & -1 & 1 \\ 1 & 1 & 1 & 1 & -1 \\ 1 & 1 & 1 & 1 & 1 \\ -1 & 1 & 1 & 1 & 1 \\ -1 & 1 & 1 & 1 & -1 \\ -1 & 1 & 1 & -1 & 1 \\ -1 & 1 & 1 & -1 & -1 \\ -1 & 1 & -1 & 1 & 1 \\ -1 & 1 & -1 & 1 & -1 \\ -1 & 1 & -1 & -1 & 1 \\ -1 & -1 & 1 & 1 & 1 \end{bmatrix}$$

Choose

$$\underline{b}^t(0) = [1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1]$$

then

$$\underline{w}(0) = \frac{1}{2^{n-1}} \underline{A}^t \underline{B}(0) = \frac{1}{16} \begin{bmatrix} 0 \\ 12 \\ 4 \\ 4 \\ 4 \end{bmatrix} = \begin{bmatrix} 0 \\ .75 \\ .25 \\ .25 \\ .25 \end{bmatrix} .$$

Since $\underline{A}\ \underline{w}(0) \not\geq \underline{0}$, one has to determine $\underline{y}(0)$, $\underline{S}(0)$, $\underline{h}(0)$, and $p(0)$. Since $\underline{y}(0) \not\leq \underline{0}$, one can proceed with the algorithm, to calculate $\underline{w}(1)$. Let $p(k)$ in equation (2.26) be used,

$$p(0) = \frac{1}{\cosh y_{max}(0)} = \frac{1}{\cosh 1} = \frac{1}{1.54305}$$

where

$$y_{max}(0) = \underset{i}{\text{Max}} \left| y_i(0) \right| = 1 .$$

$$\underline{h}^t(0) = \underline{s}^t(0) + \left| \underline{s}^t(0) \right|$$

$$= [0,0,0,0,0,0,0,1.04218,1.04218,0,0,0,0,0,0,0]$$

The algorithm terminates after the first iteration where $\underline{A}\ \underline{w}(1) > \underline{0}$ with

$$\underline{w}(1) = \frac{1}{16}\begin{bmatrix} 0 \\ 12 \\ 4 \\ 4 \\ 4 \end{bmatrix} + \frac{1}{16}\begin{bmatrix} 0 \\ .67546 \\ .67546 \\ .67546 \\ .67546 \end{bmatrix} = \frac{1}{16}\begin{bmatrix} 0 \\ 12.67546 \\ 4.67546 \\ 4.67546 \\ 4.67546 \end{bmatrix}$$

This is a desired weight vector $\underline{w}$ to be used in the switching function $g(\underline{x}) = \underline{x}^t\underline{w}$.

3. Example 3: Winder's problem of eight binary variables[6]

Consider a Boolean function of eight binary variables which correspond to the separation of two classes:

$$\text{Class 1} = \{27 \text{ to } 31,\ 39, 41, \text{to } 47,\ 49 \text{ to } 63,\ 71, 73 \text{ to } 79$$
$$81 \text{ to } 127,\ 131,\ 133 \text{ to } 255\}$$
$$= \{{}_j\underline{x}_1\}\ ,\ (j=1,2,\ldots,207)$$

$$\text{Class 2} = \{0 \text{ to } 26,\ 32 \text{ to } 38,\ 40, 48, 64 \text{ to } 70,\ 72,\ 80,$$
$$128 \text{ to } 130,\ 133\}$$
$$= \{{}_j\underline{x}_2\}\ (j=208,\ldots,256)$$

Here n=9 and m=256. For

$$\underline{b}^t(0) = [1,1,1,\ldots 1,1,1].$$

and p(k) given in equation (2.26), the algorithm terminates after the 4th iteration and gives a solution weight vector $\underline{w}_3$ for the switching function $g(\underline{x}) = \underline{x}^t\underline{w}$,

$$\underline{w}_3 = \begin{bmatrix} 1.0077 \\ 0.6136 \\ 0.4694 \\ 0.4694 \\ 0.3508 \\ 0.3508 \\ 0.1704 \\ 0.1405 \\ 0.1405 \end{bmatrix}$$

4. Example 4: A switching function of six binary variables

Consider a Boolean function of six binary variables which correspond to the separation of two classes:

$$\text{Class } C_1 = \{30,31,41 \text{ to } 63\} = \{{}_j\underline{x}_1\} \quad (j=1,\ldots,25)$$

$$\text{Class } C_2 = \{0 \text{ to } 29,\ 32 \text{ to } 40\} = \{{}_j\underline{x}_2\} \quad (j=26,\ldots,64)$$

Here n=7 and m=64. For

$$\underline{b}^t(0) = [.1,.1,.1,\ldots.1,.1,.1]$$

and p(k) given in equation (2.47), the algorithm terminates after the 1st iteration and gives a solution weight vector $\underline{w}_4$ for the switching function $g(\underline{x}) = \underline{x}^t\underline{w}$,

$$\underline{w}_4 = \begin{bmatrix} -0.8287 \\ 1.9149 \\ 1.2763 \\ 0.9954 \\ 0.3425 \\ 0.3425 \\ 0.1246 \end{bmatrix}$$

5. <u>Example 5</u>: Another switching problem of six binary variables

Consider a Boolean function of six binary variables which corresponds to the separation of two classes:

$$\text{Class } C_1 = \{46,47,53 \text{ to } 63\} = \{{}_j\underline{x}_1\},\ (j=1,2,\ldots,15)$$

$$\text{Class } C_2 = \{0 \text{ to } 45,\ 48 \text{ to } 52\} = \{{}_j\underline{x}_2\},\ (j=16,\ldots.,64)$$

Here n=7 and m=64. For

$$\underline{b}^t(0) = [1,1,1\ldots,1,1,1]$$

and p(k) given in equation (2.26), the algorithm terminates after the 1st iteration and gives a solution weight vector $\underline{w}_5$ for the switching function $g(\underline{x}) = \underline{x}^t\underline{w}$,

$$\underline{w}_5 = \begin{bmatrix} -0.7598 \\ 0.5723 \\ 0.4219 \\ 0.3281 \\ 0.2344 \\ 0.1406 \\ 0.0469 \end{bmatrix}$$

6. Example 6: Another switching problem of eight binary variables

Consider a Boolean function of eight binary variables which corresponds to the separation of two classes:

$$\text{Class } C_1 = \{127, 191, 215, 217 \text{ to } 255\} = \{{}_j\underline{x}_1\} \quad (j=1,\ldots,42)$$

$$\text{Class } C_2 = \{0 \text{ to } 126,\ 128 \text{ to } 190,\ 192 \text{ to } 214,\ 216\}$$
$$= \{{}_j\underline{x}_2\} \quad (j=43,\ldots,256)$$

Here n=9 and m=256. For

$$\underline{b}^t(0) = [.1,.1,.1,\ldots,.1,.1,.1]$$

and p(k) given in equation (2.47), the algorithm terminates after the 10th iteration and gives a solution weight vector $\underline{w}_6$ for the switching function $g(\underline{x}) = \underline{x}^t\underline{w}$,

$$\underline{w}_6 = \begin{bmatrix} 0.3732 \\ 0.2278 \\ 0.2278 \\ 0.1654 \\ 0.0769 \\ 0.0569 \\ 0.0247 \\ 0.0247 \\ 0.0247 \end{bmatrix}$$

7. Example 7: A nonlinearly separable problem of eight binary variables

Consider the following two classes of vertices of an eight-dimensional hypercube:

Class C_1 = {5 to 11,20,21,27,28,35,36,44,51,60,76,91,92,106, 107,121,122,136,137,151,152,167,182,183,197,198, 212,123,227,228,243 to 252} = $\{{}_j\underline{x}_1\}$ (j=1,...,46)

Class C_2 = {0 to 4, 12 to 19, 22 to 26, 29 to 34, 37 to 43, 45 to 50, 52 to 59, 61 to 75, 77 to 90, 93 to 105, 108 to 120, 123 to 135, 138 to 150, 153 to 166, 168 to 181, 184 to 196, 199 to 211, 214 to 226, 229 to 242, 253 to 255} = $\{{}_j\underline{x}_2\}$ (j=47,...,256)

Here n=9 and m=256. For

$$\underline{b}^t(0) = [.1,.1,.1,\ldots,.1,.1,.1]$$

and p(k) given in equation (2.47), after the zeroth iteration, the algorithm gives $\underline{y}(0) \leq \underline{0}$ which indicates that the given sets of vertices are not linearly separable.

8. Discussion

The last five example problems have been solved by the use of the proposed algorithm with various values of $\underline{b}(0)$ and with either $p(k) = \frac{1}{\cosh y_{max}(k)}$ as given by (2.26) or p(k) given by (2.47). In all cases, $\underline{b}(0)$ has equal components, i.e, $b_1(0) = b_2(0) = \ldots = b_m(0)$. The numbers of iterations required to solve the example problems in all experiments are shown in Table 1 and Table 2. These example problems have also been solved using the Ho-Kashyap algorithm, and the results are shown in Table 3 and Table 4. Note that in each of these examples with the Ho-Kashyap algorithm the number of iterations required does not change for different initial values of the $\underline{b}(0)$ vector. But the number of iterations required does change for different initial values of the $\underline{b}(0)$ vector with the proposed algorithm, as shown in Table 1. This is so because $\underline{b}(0)$ influences p(k). Also note that the number of iterations required for the proposed algorithm with $p(k) = \frac{1}{\cosh y_{max}(k)}$ and

Table 1. Number of iterations required to solve the example problems using the proposed algorithm with

$$p(k) = \frac{1}{\cosh y_{max}(k)} .$$

Example No. $b_i(0)$	3	4	5	6	7*
2.0	9	112	2		0
1.0	4	42	1		
0.5	3	29	1		0
0.2	3	26	1	231	0
0.1	3	25	1	229	
0.05	3	25	1	229	
0.01	3	25	1	229	
0.001	3	25	1	229	
10^{-4}	3	25	1	230	
10^{-5}	3	25	1	245	
10^{-6}	3	29	1	340	
10^{-7}	5	32	1		

*Not linearly separable.

Table 2. Number of iterations required to solve the example problems using the proposed algorithm with p(k) given by equation (2.47).

Example No. $b_i(0)$	3	4	5	6	7*
2.0					
1.0					
0.5					
0.2					
0.1	2	1	1	10	0
0.05					
0.01	2	1	1	10	
0.001					
10^{-4}					
10^{-5}					
10^{-6}					
10^{-7}					

*Not linearly separable.

Table 3. Number of iterations required to solve the example problems using the Ho-Kashyap algorithm with p=0.5.

Example No.	3	4	5	6	7*
$b_i(0)$					
2.0	5	52	1		0
1.0	5	52	1		
0.5	5	52	1		0
0.2	5	52	1		0
0.1	5	52	1	462	
0.05	5	52	1	462	
0.01	5	52	1	462	
0.001	5	52	1	462	
10^{-4}	5	52	1		
10^{-5}	5	52	1		
10^{-6}	5	52			
10^{-7}	5	52	1		

*Not linearly separable.

Table 4. Number of iterations required to solve the example problems using the Ho-Kashyap algorithm with p=1.0.

Example No.	3	4	5	6	7*
$b_i(0)$					
2.0	3	25	1		
1.0	3	25	1		
0.5	3	25	1		0
0.2	3	25	1		0
0.1	3	25	1	229	
0.05	3	25	1	229	
0.01	3	25	1	229	
0.001	3	25	1	229	
10^{-4}	3	25	1	229	
10^{-5}	3	25	1	229	
10^{-6}	3	25	1	229	
10^{-7}	3	25	1		

*Not linearly separable.

$0.5 \geq b_i(0) \geq 0.001$ is less than that for the Ho-Kashyap algorithm with $p = 0.5$, and is equal to that for the Ho-Kashyap algorithm with $p = 1.0$. The value of $p = 1.0$ for the Ho-Kashyap algorithm minimizes the number of iterations required for switching functions[13]. For extremely small $b_i(0)$, $b_i(0) \leq 10^{-4}$, as well as larger $b_i(0)$, $b_i(0) > 0.1$, the proposed algorithm with $p(k) = \frac{1}{\cosh y_{max}(k)}$ may take more iterations, for the proposed algorithm with the optimum p(k) given by (2.47), the number of iterations required is less than or equal to that of the Ho-Kashyap algorithm with $p = 1.0$. In the problems where the Ho-Kashyap algorithm required a very large number of iterations, the proposed algorithm reduced this number by a fairly large factor.

It has been observed in these experiments that the proposed algorithm reduced the computing time also. For example, for problems requiring a few iterations for the Ho-Kashyap algorithm the total computing time was reduced from 90 seconds to 19 seconds and execution time reduced from 30 seconds to 10 seconds with a dollar saving of $4.00, from $5.00 to $1.00. For problems requiring a large number of iterations for the Ho-Kashyap algorithm the proposed algorithm reduced the total computing time from 80 minutes to 50 seconds and execution time from 30 minutes to 5 seconds with a cost reduction of $22.00, from $23.50 to $1.50.

For a given problem, different initial values of the $\underline{b}(0)$ vector lead to different solution weight vectors, $\underline{w}$.

It has also been observed that if two $\underline{b}(0)$ vectors differed by a constant factor, the solution weight vectors thus obtained also differed by the same factor as long as the number of iterations required remained the same.

Let the number of elements of $\underline{A}\,\underline{w}(k)$ that are less than zero be designated as an error index for a set of linear inequalities at the k^{th} iteration step. This error index is represented by the number of ${}_i\underline{A}\,\underline{w}$ that are less than zero, where ${}_i\underline{A}$ is the i^{th} row of the matrix $\underline{A}$. Table 5 shows the number of ${}_i\underline{A}\,\underline{w} > 0$ observed in the experiments for examples 3, 4, 5, and 6 using both the Ho-Kashyap algorithm and the proposed algorithm with $\underline{b}^t(0) = [0.1,0.1,\ldots,0.1]$. The sum of ${}_i\underline{A}\,\underline{w} > 0$ and ${}_i\underline{A}\,\underline{w} < 0$ equals 2^m which for examples 3, 4, 5, and 6 equal 256, 64, 64, and 256 respectively. Note that, after the zeroth iteration, this error index for the proposed algorithm with $p(k) = \frac{1}{\cosh y_{max}(k)}$ is less than or equal to the error index for the Ho-Kashyap algorithm with p=0.5 and is equal to that for the Ho-Kashyap with p=1.0. The error index for the proposed algorithm with p(k) given by (2.47) is always less than or equal to that for the Ho-Kashyap algorithm with p=1.0. This error information assures the effectiveness of the proposed algorithm.

For the algorithm developed there is no guarantee that all $w_i > 0$, (i=1,...,n), which is necessary for a threshold logic circuit realizable by transistors. Since there is no prior knowledge about a Boolean function, one does not know if it is linearly separable by a weight vector with all positive elements.

Table 5. Comparison of the Error Indices for the Proposed Algorithm and the Ho-Kashyap Algorithm with $b_i(0)=0.1$ for all i.

Example	Iteration No.	Ho-Kashyap $p=0.5$ No. of $(_i\underline{Aw}>0)$	Ho-Kashyap $p=1.0$ No. of $(_i\underline{Aw}>0)$	Proposed p(k) given by Eq.(2.26) No. of $(_i\underline{Aw}>0)$	Proposed p(k) given by Eq.(2.47) No. of $(_i\underline{Aw}>0)$
3	0	241	241	241	241
	1	250	254	254	242
	2	250	254	254	256
	3	254	256	256	
	4	254			
	5	256			
4	0	60	60	60	60
	1	62	63	63	64
	2	62	63	63	
	3	63	63	63	
	...	...	...	...	
	24	63	63	63	
	25	63	64	64	
	...	...			
	51	63			
	52	64			
5	0	62	62	62	62
	1	64	64	64	64
6	0	242	242	242	242
	1	246	250	250	250
	2	250	250	250	250
	3	250	250	250	250
	4	250	250	250	250
	5	250	250	250	254
	6	250	250	250	254
	7	250	250	250	254
	8	250	250	250	254
	9	250	250	250	254
	10	250	250	250	256
	11	250	250	250	
	...	...	...	...	
	36	250	250	250	
	37	250	251	251	
	...	...	...	...	
	43	250	251	251	
	44	250	252	252	
		...	...	...	

Table 5. (Continued)

Example	Iteration No.	Ho-Kashyap $p=0.5$ No. of $(_i\underline{Aw}>0)$	Ho-Kashyap $p=1.0$ No. of $(_i\underline{Aw}>0)$	Proposed p(k) given by Eq.(2.26) No. of $(_i\underline{Aw}>0)$	Proposed p(k) given by Eq.(2.47) No. of $(_i\underline{Aw}>0)$
	74	250	252	252	
	75	251	252	252	
	...	...	...	...	
	90	251	252	252	
6	91	252	252	252	
	...	...	...	...	
	228	252	252	252	
	229	252	256	256	
	...	...	...	...	
	461	252			
	462	256			

IV. APPLICATION OF THE ACCELERATED ALGORITHM TO PATTERN RECOGNITION

For dichotomization of patterns other than switching function problems, no simplification of the algorithm can be made, and the proposed algorithm in equation (2.29) together with p(k) given in equation (2.26) or equation (2.47) will be used. The generalized inverse of the matrix $\underline{A}$ must be calculated once per problem for the abstraction aspect in pattern recognition. When a solution weight vector $\underline{w}$ is obtained from the application of the algorithm, it can be used in the pattern recognizer as illustrated in Figure 3. A digital computer program for the algorithm (2.29) has been written in MAD language. The calculation of $A^{\#}$ was obtained according to Kalman and Englar's scheme[22]. The program was originally written in FORTRAN and then translated into MAD language to be consistent with the language used for the proposed algorithm. The complete computer program is included in Appendix B. The proposed algorithm has been applied to the two pattern classification problems as described in the next two sections.

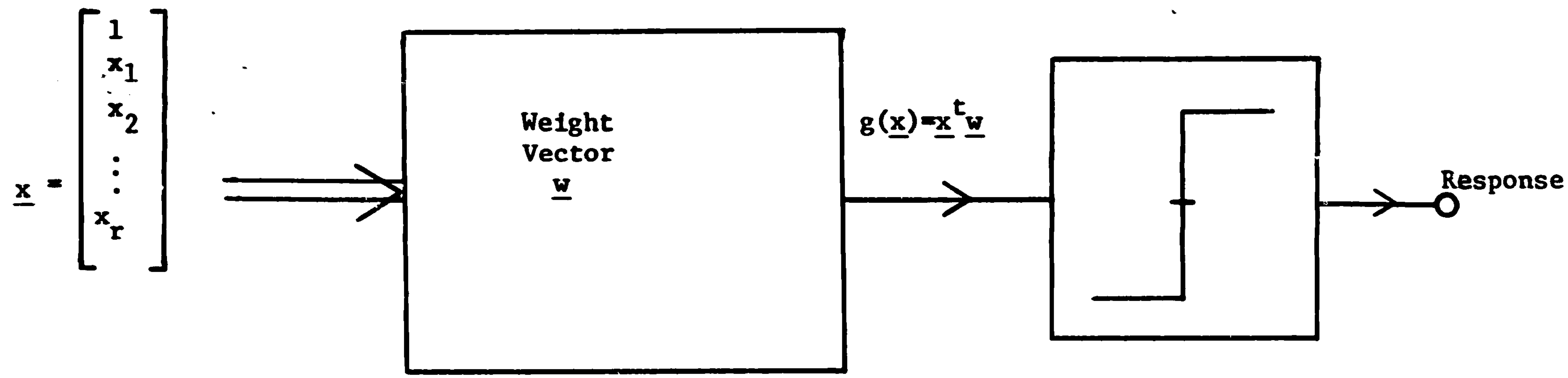


Figure 3. Block Diagram for a Pattern Dichotomizer.

A. A Character Recognition Problem

In this study patterns consisting of four pairs of hand printed alphanumeric characters were considered. The data were obtained from the Learning Research and Development Center at the University of Pittsburgh. One of the Center's activities is to teach children, ages five to eight, the alphabet and numbers, via instructional devices and computers. Hence, the machine recognition of hand printed characters is a current research interest in the Center. The selected pairs of characters are similar in form and the patterns collected are representative of children's hand printing. The four pairs considered here are A and H, Z and 2, I and 1, and G and 6. Each character was written inside a square with 12 by 12 divisions. Five attributes or pattern components, x_o, x_1, x_2, x_3, and x_4, were obtained from each pair of characters for classification. The first attribute was the height of the character and was normalized to be 1.0. The other attributes were certain length and width, etc., each of which was a fraction of this height. The attributes given to describe the four pattern pairs are shown in Figure 4 to Figure 7. These represent sets of crude but simple features of hand printed character pairs. The pattern components of character pairs for the sample or training sets are listed in Table 6. The original hand printed characters are reproduced in Appendix C. Note that since the normalized height is unity for all characters, it can be assigned as the x_o component or threshold attribute of the $\underline{x}$ vector. Hence $\underline{x}$ is a 5 by 1 vector with n = 5.

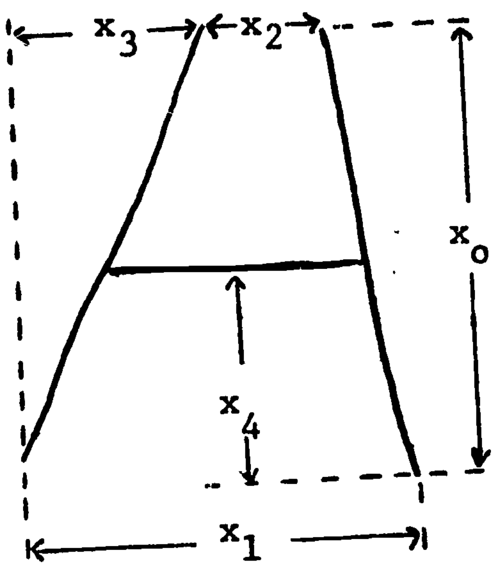


Figure 4. Pattern Components of the A-H Pair.

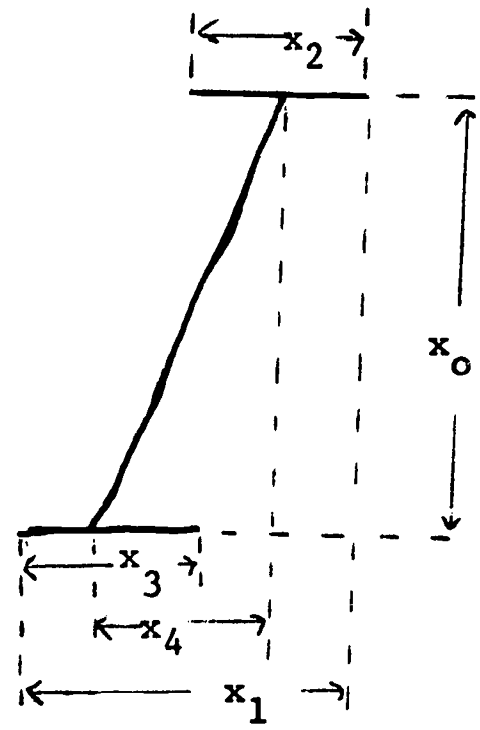


Figure 5. Pattern Components of the I-1 Pair.

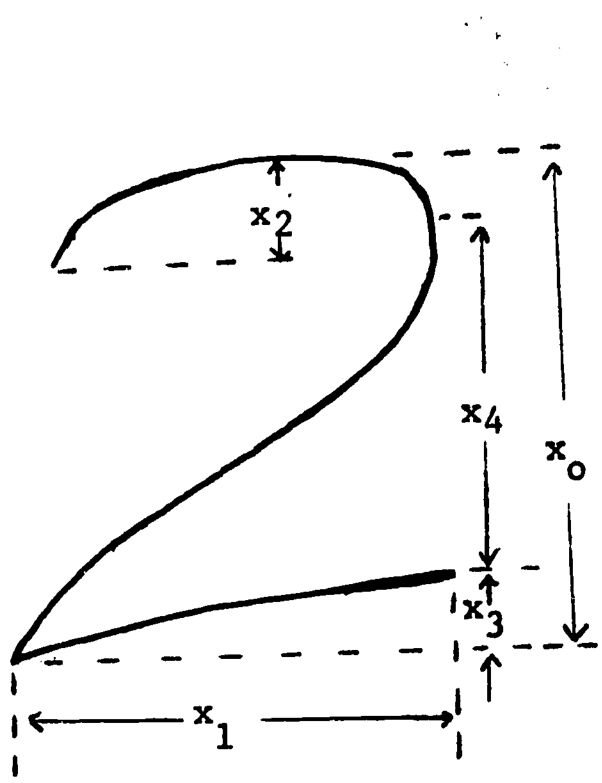


Figure 6. Pattern Components of the Z-2 Pair

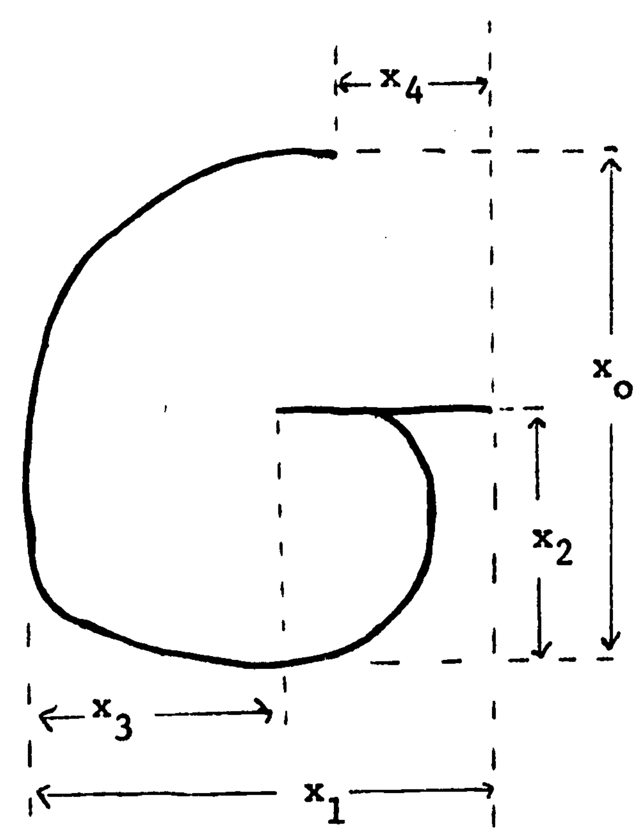


Figure 7. Pattern Components of the G-6 Pair.

Table 6. Pattern Components of the Character Pairs Used in the Training Sets.

A

x_o	x_1	x_2	x_3	x_4
1.0	.8	.0	.5	.5
1.0	.8	.0	.5	.25
1.0	.8	.0	.5	.35
1.0	.8	.0	.5	.75

H

x_o	x_1	x_2	x_3	x_4
1.0	.6	.6	.0	.5
1.0	.6	.5	.08	.5
1.0	.6	.4	.16	.5
1.0	.6	.3	.25	.5
1.0	.6	.2	.3	.5
1.0	.6	.6	.0	.3
1.0	.6	.6	.0	.6

Z

x_o	x_1	x_2	x_3	x_4
1.0	.7	.08	.08	1.0
1.0	.7	.17	.08	1.0
1.0	.7	.25	.08	1.0
1.0	.7	.42	.08	1.0
1.0	1.0	.08	.08	1.0
1.0	1.1	.08	.08	1.0
1.0	1.33	.08	.08	1.0

2

x_o	x_1	x_2	x_3	x_4
1.0	.6	.2	.08	.7
1.0	.6	.1	.08	.7
1.0	.6	.3	.08	.7
1.0	.6	.08	.08	.7
1.0	.6	.1	.08	.5
1.0	.6	.1	.08	.3
1.0	.6	.1	.08	.8

I

x_o	x_1	x_2	x_3	x_4
1.0	.3	.3	.3	.0
1.0	.4	.25	.4	.1
1.0	.4	.3	.25	.15
1.0	.7	.45	.5	.35
1.0	.5	.3	.25	.25

1

x_o	x_1	x_2	x_3	x_4
1.0	.05	.0	.0	.05
1.0	.2	.0	.0	.2
1.0	.1	.0	.0	.1
1.0	.5	.0	.0	.5
1.0	.4	.0	.0	.4

G

x_o	x_1	x_2	x_3	x_4
1.0	.8	.5	.8	.0
1.0	.8	.5	.2	.0
1.0	.8	.5	.5	.0
1.0	.8	.5	.62	.0
1.0	.8	.5	.8	.25
1.0	.8	.5	.8	.12
1.0	.8	.5	.8	.2

6

x_o	x_1	x_2	x_3	x_4
1.0	.5	.5	.0	.33
1.0	.5	.3	.0	.25
1.0	.5	.4	.0	.33

It was desired to recognize A against H, Z against 2, I against 1, G against 6 , or vice versa. For each pair group, designate patterns of one character as belonging to class C_1 and patterns of the other character class C_2. For example, character A belongs to class C_1 in the first pair, character Z belongs to class C_1 in the second pair, character I belongs to class C_1 in the third pair, and character G belongs to class C1 in the fourth pair. For each pair group, a discriminant function, $g(\underline{x}) = \underline{x}^t \underline{w}$, was to be determined. As shown in Table 6, there were eleven sample patterns for the A-H pair, fourteen sample patterns for the Z-2 pair, and ten sample patterns each for the I-1 and G-6 pairs. The size of matrix $\underline{A}$ varied from 10 by 5 to 14 by 5. The proposed algorithm was applied to each pair group with $\underline{b}^t(0) = [0.1, 0.1, \ldots, 0.1]$ and p(k) given by equation (2.47) to obtain the following solution weight vectors, $\underline{w}_{AH}$, $\underline{w}_{Z2}$, $\underline{w}_{I1}$, and $\underline{w}_{G6}$:

$$\underline{w}_{AH} = \begin{bmatrix} .0070 \\ .0100 \\ .0001 \\ -.0001 \\ .0001 \end{bmatrix}$$

$$\underline{w}_{Z2} = \begin{bmatrix} .0037 \\ .0014 \\ .0010 \\ -.0003 \\ .0031 \end{bmatrix}$$

$$\underline{w}_{I1} = \begin{bmatrix} .0851 \\ .0001 \\ .1566 \\ .1743 \\ -.0001 \end{bmatrix}$$

$$\underline{w}_{G6} = \begin{bmatrix} .4333 \\ .6661 \\ .0001 \\ .0001 \\ -.0001 \end{bmatrix}$$

where the first subscript refers to class C_1 and the second subscript refers to class C_2 in each pair group. These solution weight vectors were all obtained after the zeroth iteration.

The solution weight vectors were also tested by some new sample patterns. Only in the Z-2 pair group, there was one misclassification among a total of twelve new sample patterns. This misclassified one was a 2 which was written so ambiguously that even a human observer could hardly distinguish it from Z.

B. A Biomedical Pattern Recognition Problem

The proposed algorithm was also applied to a biomedical pattern recognition problem. The problem is to investigate whether or not a change exists in the diurnal cycle of an individual person upon a change in his environmental condition or physiological state and if such a change may be used to diagnose physical ailements under strictly controlled conditions by measuirng the amounts of electrolytes present in urine and blood samples every three hours. The problem and data were presented by Dr. Venucci of the School of Medicine, University of Pittsburgh. The data consisted of thirteen sample patterns under two different conditions. Each pattern has eight components which represent the concentrations of electrolytes. Thus $N = 13$ and $n = r+1 = 8+1 = 9$; the size of the pattern matrix $\underline{A}$ is 13 by 9. The pattern matrix $\underline{A}$ is shown in Table 7. Let $\underline{b}^t(0) = [0.1,0.1,\ldots,0.1]$. For this problem the Ho-Kashyap algorithm with $p = 1$ required 927 iterations to determine the separability. However,

Table 7. The Pattern Matrix A from a Biomedical Experiment

x_o	x_1	x_2	x_3	x_4	x_5	x_6	x_7	x_8
1.00	.96	1.19	1.35	.75	1.12	.94	.73	.97
1.00	.75	1.19	1.35	1.06	1.07	.97	.81	.81
1.00	.80	1.13	.85	.90	1.14	1.27	1.01	.88
1.00	.66	1.40	1.25	1.09	1.54	.79	.27	.00
1.00	2.04	1.14	1.10	.57	.62	.66	.47	1.39
1.00	1.02	1.32	1.06	1.03	1.07	1.16	.77	.57
-1.00	-.48	-1.01	-.68	-.72	-1.76	-1.25	-.62	-1.47
-1.00	-.55	-.55	-1.04	-.91	-1.40	-1.17	-1.28	-1.09
-1.00	-.87	-.79	-1.34	-.86	-.44	-2.15	-.82	-.74
-1.00	-.09	-.70	-.67	-.80	-1.93	-1.29	-1.14	-1.39
-1.00	-1.12	-1.75	-.51	-.72	-1.25	-.46	-.89	-1.29
-1.00	-1.20	-1.47	-.60	-.96	-1.13	-.89	-.74	-1.00
-1.00	-1.43	-1.79	-.68	-.75	-.82	-.56	-.94	-1.04

the proposed algorithm with p(k) given by equation (2.47) required only two iterations, where p(0) = 5.270684 and p(1) = 3.197152. The problem is linearly separable and a solution weight vector $\underline{w}$ obtained by the proposed algorithm is

$$\underline{w} = \underline{w}(2) = \begin{bmatrix} 13.6089 \\ 2.5915 \\ 1.6847 \\ 2.2314 \\ 0.3414 \\ 3.0077 \\ 1.8428 \\ 1.6559 \\ 0.0096 \end{bmatrix} .$$

It was observed in this case that the proposed algorithm reduced the number of iterations required by a factor of approximately 450 over that required for the Ho-Kashyap algorithm.

Notice that in the examples above components of a weight vector for a given pattern may differ in magnitude by as much as 1700. Although the magnitude of the attributes differ by as much as 200 it is possible that some of the attributes are not necessary to describe the pattern.

This result is determined by noticing the small effect of the products of these attributes and their corresponding weights have on the inequality. For economic reasons one would choose the least number of attributes to describe a pattern, but for flexability and reliability it is necessary to have sufficient attributes. This suggests the development of an experimental procedure to select an adequate set of attributes.

V. GENERALIZATION OF THE ACCELERATED ALGORITHM OF LINEAR INEQUALITIES TO MULTICLASS PATTERN CLASSIFICATION

A. Preliminary Remarks

The problem of multiclass patterns classification is that it must be determined to which of the R different classes, $C_1, C_2, \ldots, C_R$, a given pattern vector, $\underline{x}$, belongs. If the R-class patterns are linearly separable, there exist R weight vectors $\underline{w}_j$ to construct R discriminant functions $g_j(\underline{x})$, $(j=1,2,\ldots,R)$, such that

$$g_j(\underline{x}) = \underline{x}^t \underline{w}_j > \underline{x}^t \underline{w}_i = g_i(\underline{x}) \text{ for all } i \neq j,\ \underline{x} \in C_j \qquad (5.1)$$

The improved algorithm for dichotomization obtained in Chapter II will be generalized to the multiclass pattern classification. A similar criterion function will be specified and a convergent iterative algorithm will be devised, incorporating the gradient descent procedure, to make the proposed multiclass algorithm a direct analog of the previously described dichotomous algorithm.

The notion of equilateral simplex will be used[14-17]. Chaplin and Levadi[14] have formulated another set of inequalities, other than (5.1), which can be considered as the representation of linear separation of R-class patterns. This set of inequalities is

$$||\underline{x}^t\underline{U} - \underline{e}^t_j|| < ||\underline{x}^t\underline{U} - \underline{e}^t_i|| \text{ for all } i \neq j,\ \underline{x} \in C_j \quad \text{for all } j=1,2,\ldots,R \tag{5.2}$$

where $\underline{U}$ is an n by(R-1) weight matrix and the vectors $\underline{e}_j$'s are the vertex vectors of a R-1 dimensional equilateral simplex with its centroid at the origin. If each $\underline{e}_j$ is associated with one class, $\underline{x}$ is classified according to the nearest neighborhood of the mapping $\underline{U}^t\underline{x}$, as illustrated in Figure 8. The (R-1) by 1 vectors $\underline{e}_j$'s have the following properties:

$$\begin{aligned} &||e_j|| = 1 \quad \text{for all } j=1,2,\ldots,R \\ &||\underline{e}_j - \underline{e}_i|| = ||e_j - e_k|| \text{ for all } i,\ k \neq j \end{aligned} \tag{5.3}$$

and

$$\underline{e}^t_j\,(\underline{e}_j - \underline{e}_i) > 0 \quad \text{for all } i \neq j\ . \tag{5.4}$$

The components of $\underline{e}^t_j$, $\underline{e}^t_j = [e_{j1},\ldots e_{ji},\ldots e_{j(R-1)}]$, are determined as follows:

$$\underline{e}_{ji} = \begin{cases} [(\frac{R}{R-1})(\frac{R-i}{R-i+1})]^{1/2} & \text{for } j=i \\ -\frac{1}{(R-i)}\,[(\frac{R}{R-1})(\frac{R-i}{R-i+1})]^{1/2} & \text{for } j > i \\ 0 & \text{for } j < i \end{cases} \tag{5.5}$$

$(j=1,2,\ldots,R;\ i=1,2,\ldots,R-1)$.

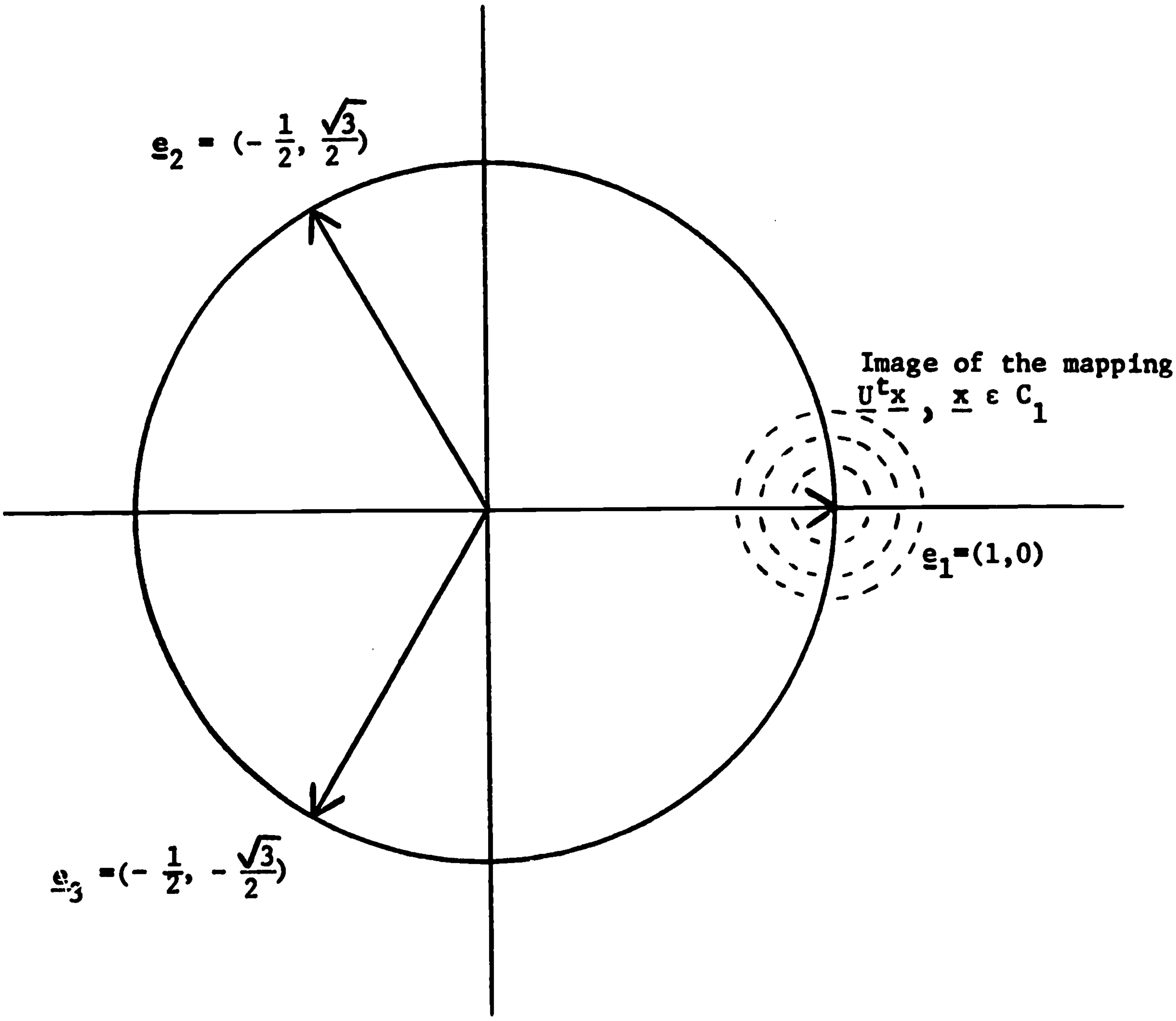


Figure 8. Equilateral Simplex Vertices and Nearest Neighborhood Mapping of Pattern Vectors , R = 3.

Inequalities (5.2) are, in fact, equivalent to inequalities (5.1). This will be shown below. Rewriting inequalities (5.2) one obtains

$$\{ (\underline{U}^t\underline{x} - \underline{e}_j)^t (\underline{U}^t\underline{x} - \underline{e}_j\} < \{(\underline{U}^t\underline{x} - \underline{e}_i)^t(\underline{U}^t\underline{x} - e_i \}$$

or

$$\{ \underline{x}^t\underline{U}\,\underline{U}^t\underline{x} - \underline{x}^t\underline{U}\,\underline{e}_j - \underline{e}_j{}^t\underline{U}^t\underline{x} + \underline{e}^t{}_j\,\underline{e}_j\}$$

$$< \{\underline{x}^t\underline{U}\,\underline{U}^t\underline{x} - \underline{x}^t\underline{U}\,\underline{e}_i - \underline{e}^t{}_i\,\underline{U}^t\underline{x} + \underline{e}^t{}_i\,\underline{e}_i \},$$

$$\text{for all } i \neq j,\ x \in C_j \tag{5.6}$$

Since, from (5.3),

$$\underline{e}^t{}_j\,\underline{e}_j = 1 = \underline{e}^t{}_i\underline{e}_i$$

equation (5.6), upon simplification, reduces to

$$-\underline{x}^t\underline{U}(\underline{e}_j-\underline{e}_i) < - (\underline{e}^t{}_i-\underline{e}^t{}_j)\underline{U}^t\underline{x},\quad \text{for all } i \neq j,\ \underline{x} \in C_j$$

$$\underline{x}^t\underline{U}(\underline{e}_j-\underline{e}_i) > - [\underline{x}^t\underline{U}(\underline{e}_j-\underline{e}_i)]^t,\ \text{for all } i \neq j\ \ \underline{x} \in C_j \tag{5.7}$$

Since $\underline{x}^t\underline{U}(\underline{e}_j-\underline{e}_i)$ is a scalar, the above inequality implies

$$\underline{x}^t\underline{U}(\underline{e}_j - \underline{e}_i) > 0 \text{ for all } i \neq j\ \ \underline{x} \in C_j\,. \tag{5.8}$$

Let

$$\underline{w}_j = \underline{U}\,\underline{e}_j,\quad j=1,2,\ldots,R \tag{5.9}$$

Then (5.8) becomes

$$\underline{x}^t\underline{w}_j > \underline{x}^t\underline{w}_i \quad \text{for all } i \neq j, \ \underline{x} \in C_j$$

which is (5.1).

However, in order to generalize the dichotomous algorithm of Chapter II to a multiclass algorithm, additional information of linear inequalities is necessary[15]. Let the N x n pattern matrix $\underline{A}$ be defined in the following manner.

$$\underline{A} = \begin{bmatrix} \underline{A}_1 \\ \\ \hline \vdots \\ \hline \underline{A}_j \\ \\ \hline \vdots \\ \hline \\ \underline{A}_R \end{bmatrix} \triangleq \begin{bmatrix} {}_1\underline{x}^t{}_1 \\ \vdots \\ {}_{n_1}\underline{x}^t{}_1 \\ \hline \vdots \\ \hline {}_1\underline{x}^t{}_j \\ \vdots \\ {}_{n_j}\underline{x}^t{}_j \\ \hline \vdots \\ \hline {}_1\underline{x}^t{}_R \\ \vdots \\ {}_{n_R}\underline{x}^t{}_R \end{bmatrix} \tag{5.10}$$

where $\underline{A}_j$ is an n_j by n submatrix having as its rows n_j transposed pattern vectors of class C_j, ${}_i\underline{x}^t{}_j$, $(i=1,2,\ldots,n_j)$, and $N = n_1 + n_2 + \ldots + n_R$. Designate the n by (R-1) weight matrix $\underline{U}$ as composed of (R-1) column vectors $\underline{u}_q$, $(q=1,2,\ldots,R-1)$.

$$\underline{U} = \left[u_1 \;\Big|\; \cdots \;\Big|\; \underline{u}_q \;\Big|\; \cdots \;\Big|\; \underline{u}_{R-1} \right] . \tag{5.11}$$

Also define an N by(R-1) matrix $\underline{B}$ as

$$\underline{B} = \begin{bmatrix} \underline{B}_1 \\ \hline \vdots \\ \hline \underline{B}_j \\ \hline \vdots \\ \hline \underline{B}_R \end{bmatrix} \triangleq \begin{bmatrix} {}_1\underline{b}^t{}_1 \\ \vdots \\ {}_{n_1}\underline{b}^t{}_1 \\ \hline \vdots \\ \hline {}_1\underline{b}^t{}_j \\ \vdots \\ {}_{n_j}\underline{b}^t{}_j \\ \hline \vdots \\ \hline {}_1\underline{b}^t{}_R \\ \vdots \\ {}_{n_R}\underline{b}^t{}_R \end{bmatrix} \tag{5.12}$$

whose row vectors ${}_\ell\underline{b}^t{}_j$, $(j=1,2,\ldots,R;\ \ell=1,2,\ldots,n_j)$, correspond to the class groupings in the $\underline{A}$ matrix and satisfy the following inequalities

$$ {}_\ell\underline{b}^t{}_j\ (\underline{e}_j - \underline{e}_i) > 0 \quad \text{for all } i \neq j \quad \text{for all } j=1,2,..R\,. \tag{5.13}$$

$\underline{B}_j$ is a n_j by (R-1) submatrix of $\underline{B}$, j=1,2,...,R. Let an N by (R-1) matrix

$\underline{Y}$ be defined as

$$\underline{Y} \stackrel{\Delta}{=} \underline{A}\,\underline{U} - \underline{B}, \tag{5.14}$$

The representation of $\underline{Y}$ may be in the form of either an array of (R-1) column vectors, $\underline{y}_q$, (q=1,2,...,R-1).

$$\underline{Y} = \left[\underline{y}_1 \,\vdots \cdots \vdots\, \underline{y}_q \,\vdots \cdots \vdots\, \underline{y}_{R-1} \right] \tag{5.15}$$

or an array of N row vectors ${}_i\underline{Y}_j$, (j=1,2,...,R; i=1,2,...,n_j), corresponding to the class groupings in the $\underline{A}$ matrix

$$\underline{Y} = \begin{bmatrix} \underline{Y}_1 \\ \hline \vdots \\ \hline \underline{Y}_j \\ \hline \vdots \\ \hline \underline{Y}_R \end{bmatrix} = \begin{bmatrix} {}_1\underline{Y}_1 \\ \vdots \\ {}_{n_1}\underline{Y}_1 \\ \hline \vdots \\ \hline {}_1\underline{Y}_j \\ \vdots \\ {}_{n_j}\underline{Y}_j \\ \hline \vdots \\ \hline {}_1\underline{Y}_R \\ \vdots \\ {}_{n_R}\underline{Y}_R \end{bmatrix} \tag{5.16}$$

where $\underline{Y}_j$ is an n_j by (R-1) submatrix of $\underline{Y}$,

$$\underline{Y}_j = \underline{A}_j \, \underline{U} - \underline{B}_j \tag{5.17}$$

or

$$ {}_{\ell}\underline{Y}_j = {}_{\ell}\underline{x}^t{}_j \, \underline{U} - {}_{\ell}\underline{b}^t{}_j \qquad \begin{array}{l} j=1,2,\ldots,R \\ \ell=1,2,\ldots,n_j \end{array} \tag{5.18}$$

The set of linear inequalities which will be discussed in this chapter is, from (5.8),

$$\underline{A}_j\underline{U}(\underline{e}_j-\underline{e}_i) > \underline{0} \quad \begin{array}{l} \text{for all } i \neq j \\ \text{for all } j=1,2,\ldots,R \end{array} \tag{5.19}$$

Associated with it is another set of linear inequalities

$$\underline{Y}_j(\underline{e}_j-\underline{e}_i) = (\underline{A}_j\underline{U} - \underline{B}_j)(\underline{e}_j-\underline{e}_i) > \underline{0} \tag{5.20}$$

for all $i \neq j$

for all $j=1,2,\ldots,R$

or

$$ {}_{\ell}\underline{Y}_j(\underline{e}_j-\underline{e}_i) = ({}_{\ell}\underline{x}^t{}_j\underline{U} - {}_{\ell}\underline{b}^t{}_j)(\underline{e}_j-\underline{e}_i) > 0 \tag{5.21}$$

for all $i \neq j$

all $j=1,2,\ldots,R$

all $\ell=1,2,\ldots,n_j$

Since, by (5.13), $\underline{B}_j(\underline{e}_j - \underline{e}_i)$ is constrained to have positive components for all $i \neq j$, inequalities (5.20) or (5.21) implies the inequalities (5.19) and hence (5.1) or (5.2). When inequalities (5.19) are satisfied for all $i \neq j$ and for all $j=1,2,\ldots,R$, a solution weight matrix $\underline{U}$ is reached which will give linear classification of R-class patterns; that is, if

$$\underline{x}^t\underline{U}\ (\underline{e}_j - \underline{e}_i) > 0 \text{ for all } i \neq j$$

then $\underline{x}$ is classified as of class C_j. Also, if R weight vectors $\underline{w}_j$, $j=1,2,\ldots,R$, are computed from $\underline{U}$ according to (5.9), then R discriminant functions, $g_j(\underline{x}) = \underline{x}^t\underline{w}_j$, $(j=1,2,\ldots,R)$, can be obtained for use in the R-class pattern recognizer shown in Figure 9.

B. Development of the Algorithm

For the notational simplicity in the derivation of the gradient function to be developed below, let the matrices $\underline{A}$, $\underline{U}$, $\underline{B}$ and $\underline{Y}$ in equations (5.10), (5.11), (5.12), and (5.15) be represented respectively as

$$\underline{A} = \begin{bmatrix} a_{11} & a_{12} & a_{1n} \\ \cdots & \cdots & \cdots \\ a_{N1} & a_{N2} & a_{Nn} \end{bmatrix} \qquad (5.22)$$

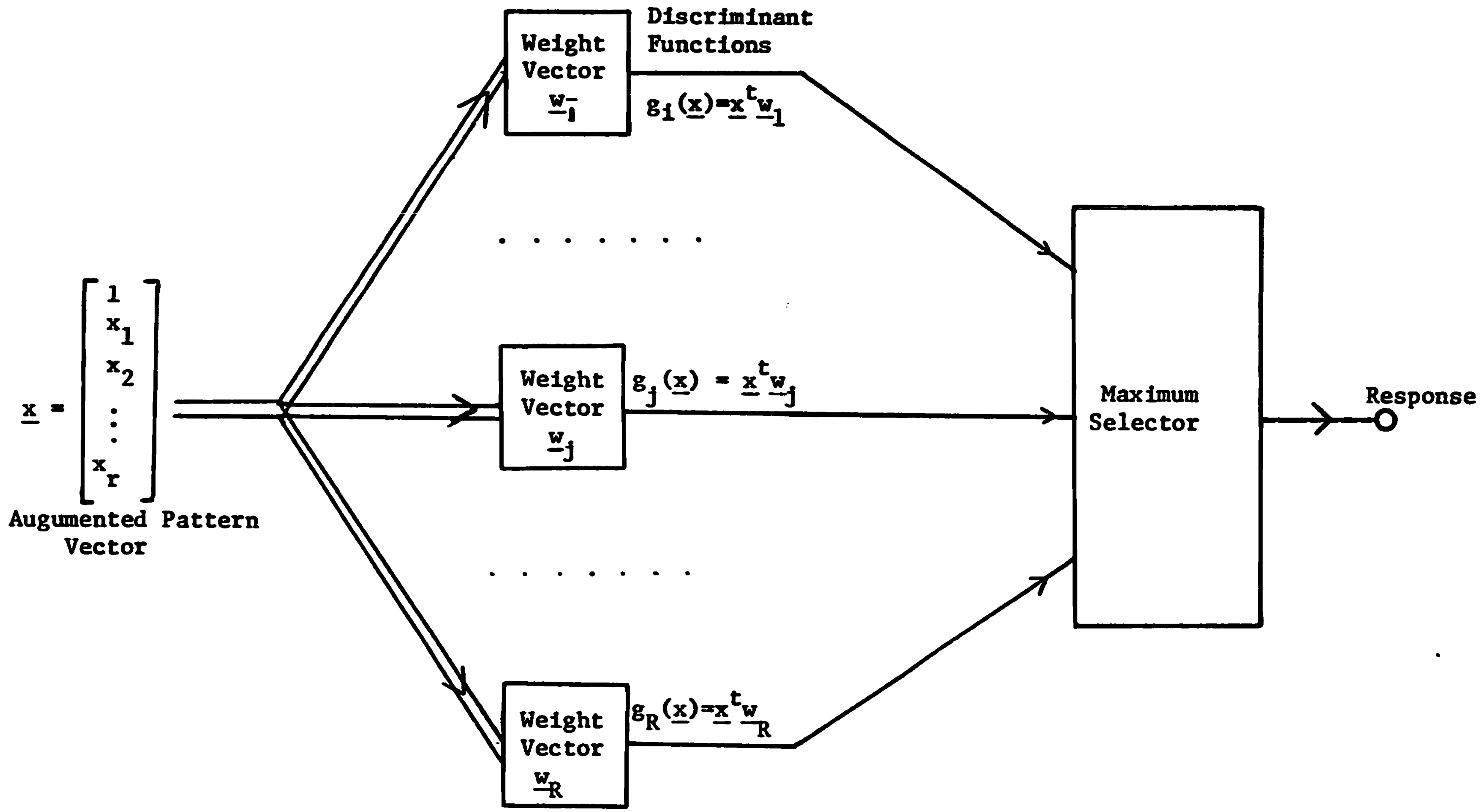


Figure 9. Block Diagram of a Multiclass Pattern Reorganizer.

$$\underline{U} = \begin{bmatrix} u_{11} & u_{12} & u_{1,R-1} \\ \cdots & \cdots & \cdots \\ u_{n1} & u_{n2} & u_{n,R-1} \end{bmatrix} \tag{5.23}$$

$$\underline{B} = \begin{bmatrix} b_{11} & b_{12} & b_{1,R-1} \\ \cdots & \cdots & \cdots \\ b_{N1} & b_{N2} & b_{N,R-1} \end{bmatrix} \tag{5.24}$$

and

$$\underline{Y} = \begin{bmatrix} y_{11} & y_{12} & y_{1,R-1} \\ \cdots & \cdots & \cdots \\ y_{N1} & y_{N2} & y_{N,R-1} \end{bmatrix} . \tag{5.25}$$

Substituting these into equation (5.14), one obtains

$$y_{ij} = \sum_{k=1}^{n} a_{ik}\, u_{kj} - b_{ij} . \tag{5.26}$$

Let $\underline{C}(\underline{Y})$ be an N by (R-1) matrix defined by

$$\underline{C}(\underline{Y}) = \begin{bmatrix} c_{11} & c_{12} & c_{1,R-1} \\ \cdots & \cdots & \cdots \\ c_{N1} & c_{N2} & c_{N,R-1} \end{bmatrix} = \begin{bmatrix} \cosh \frac{1}{2}\, y_{11} & \cosh \frac{1}{2}\, y_{12} & \cosh \frac{1}{2}\, y_{1,R-1} \\ \cdots & \cdots & \cdots \\ \cosh \frac{1}{2}\, y_{N1} & \cosh \frac{1}{2}\, y_{N2} & \cosh \frac{1}{2}\, y_{N,R-1} \end{bmatrix} . \tag{5.27}$$

The criterion function $J(\underline{Y})$ to be minimized is chosen as the trace of $4\,\underline{C}^t(\underline{Y})\underline{C}(\underline{Y})$,

$$J(\underline{Y}) \triangleq T_r(4\underline{C}^t\underline{C}) = 4\sum_{i=1}^{N}\sum_{j=1}^{R-1} c_{ij}^2 = 4\sum_{i=1}^{N}\sum_{j=1}^{R-1}(\cosh\frac{1}{2}y_{ij})^2$$

$$= \sum_{i=1}^{N}\sum_{j=1}^{R-1} J_{ij}(\underline{Y}) \tag{5.28}$$

where

$$J_{ij}(\underline{Y}) = 4(\cosh\frac{1}{2}y_{ij})^2. \tag{5.29}$$

Following the same approach of the dichotomous case, determine the gradients of $J(\underline{Y})$ with respect to both $\underline{U}$ and $\underline{B}$.

$$\frac{\partial J_{ij}(\underline{Y})}{\partial \underline{U}} = 4\,(\cosh\frac{1}{2}y_{ij}\,\sinh\frac{1}{2}y_{ij})\,\frac{\partial y_{ij}}{\partial \underline{U}}$$

$$= 2\sinh y_{ij}\,\frac{\partial y_{ij}}{\partial \underline{U}} \tag{5.30}$$

where the derivative of a scalar with respect to a matrix is a matrix. From (5.26) and (5.30),

$$\frac{\partial J_{ij}(\underline{Y})}{\underline{U}} = 2\sinh y_{ij}\begin{array}{c}\begin{bmatrix} 0 & 0 & 0 & a_{i1} & 0 & 0 \\ 0 & 0 & 0 & a_{i2} & 0 & 0 \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ 0 & 0 & 0 & a_{in} & 0 & 0 \end{bmatrix} \\ \text{jth column}\end{array} \tag{5.31}$$

Then the gradient of the criterion function, $J(\underline{Y})$, with respect to the matrix $\underline{U}$ is

$$\frac{\partial J(\underline{Y})}{\partial \underline{U}} = 2 \begin{bmatrix} \sum_{i=1}^{N} a_{i1} \sinh y_{i1} & \cdots & \sum_{i=1}^{N} a_{i1} \sinh y_{ij} & \cdots & \sum_{i=1}^{N} a_{i1} \sinh y_{i,R-1} \\ \sum_{i=1}^{N} a_{i2} \sinh y_{i1} & \cdots & \sum_{i=1}^{N} a_{i2} \sinh y_{ij} & \cdots & \sum_{i=1}^{N} a_{i2} \sinh y_{i,R-1} \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ \sum_{i=1}^{N} a_{in} \sinh y_{i1} & \cdots & \sum_{i=1}^{N} a_{in} \sinh y_{ij} & \cdots & \sum_{i=1}^{N} a_{in} \sinh y_{i,R-1} \end{bmatrix}$$

$$= 2 \begin{bmatrix} a_{11} & a_{21} & \cdots & a_{N1} \\ a_{12} & a_{22} & \cdots & a_{N2} \\ \cdots & \cdots & \cdots & \cdots \\ a_{1n} & a_{2n} & \cdots & a_{Nn} \end{bmatrix} \begin{bmatrix} \sinh y_{11} & \sinh y_{12} & \cdots & \sinh y_{1,R-1} \\ \sinh y_{21} & \sinh y_{22} & \cdots & \sinh y_{2,R-1} \\ \cdots & \cdots & \cdots & \cdots \\ \sinh y_{N1} & \sinh y_{N2} & \cdots & \sinh y_{N,R-1} \end{bmatrix}$$

$$= 2\, \underline{A}^{t}\, \underline{S}(\underline{Y}) \qquad (5.32)$$

where $\underline{S}(\underline{Y})$ is an N by (R-1) matrix with the following representation.

$$\underline{S}(\underline{Y}) = \begin{bmatrix} \sinh y_{11} & \sinh y_{12} & \cdots & \sinh y_{1,R-1} \\ \cdots & \cdots & \cdots & \cdots \\ \sinh y_{N1} & \sinh y_{N2} & \cdots & \sinh y_{N,R-1} \end{bmatrix}$$

$$= \left[\underline{s}_1(\underline{Y}) \,\middle|\, \cdots \,\middle|\, \underline{s}_q(\underline{Y}) \,\middle|\, \cdots \,\middle|\, \underline{s}_{R-1}(\underline{Y}) \right]$$

$$= \begin{bmatrix} \underline{S}_1(\underline{Y}) \\ \hline \underline{S}_j(\underline{Y}) \\ \hline \vdots \\ \hline \underline{S}_R(\underline{Y}) \end{bmatrix} = \begin{bmatrix} {}_1\underline{S}_1(\underline{Y}) \\ \vdots \\ {}_{n_1}\underline{S}_1(\underline{Y}) \\ \hline {}_1\underline{S}_j(\underline{Y}) \\ \vdots \\ {}_{n_j}\underline{S}_j(\underline{Y}) \\ \hline \vdots \\ \hline {}_1\underline{S}_R(\underline{Y}) \\ \vdots \\ {}_{n_R}\underline{S}_R(\underline{Y}) \end{bmatrix} \tag{5.33}$$

and where ${}_i\underline{S}_j(\underline{Y})$ is a row vector of the following form

$${}_i\underline{S}_j(\underline{Y}) = [\sinh y_{(n_{j-1}+i),1}, \ldots, \sinh y_{(n_{j-1}+i),R-1}], \tag{5.34}$$

From (5.29)

$$\frac{\partial J_{ij}(\underline{Y})}{\partial \underline{B}} = 4 \left(\cosh \frac{1}{2} y_{ij} \sinh \frac{1}{2} y_{ij}\right) \frac{\partial y_{ij}}{\partial \underline{B}}$$

$$= 2 \sinh y_{ij} \frac{\partial y_{ij}}{\partial \underline{B}} \tag{5.35}$$

From (5.26)

$$\frac{\partial y_{ij}}{\partial \underline{B}} = \begin{array}{c c} & \begin{array}{ccccc} & & j & & \\ & & \downarrow & & \end{array} \\ \begin{array}{c} \\ \\ i \rightarrow \\ \\ \\ \end{array} & \begin{bmatrix} 0 & \cdots & 0 & \cdots & 0 \\ 0 & \cdots & \vdots\,0 & & \vdots \\ 0 & \cdots & -1 & \cdots & 0 \\ 0 & \cdots & 0 & & \vdots \\ 0 & \cdots & \vdots\,0 & & 0 \end{bmatrix} \end{array} \tag{5.36}$$

Substituting (5.36) into (5.35) gives

$$\frac{\partial J_{ij}(\underline{Y})}{\partial \underline{B}} = -2 \begin{array}{c c} \begin{bmatrix} 0 & \cdots & 0 & \cdots & 0 \\ 0 & \cdots & 0 & \cdots & 0 \\ 0 & \cdots & \sinh y_{ij} & \cdots & 0 \\ 0 & \cdots & 0 & \cdots & 0 \\ 0 & \cdots & 0 & \cdots & 0 \end{bmatrix} & \begin{array}{c} \\ \\ \leftarrow i \\ \\ \\ \end{array} \\ \begin{array}{ccccc} & & \uparrow & & \\ & & j & & \end{array} & \end{array} \tag{5.37}$$

Hence, the gradient of the criterion function $J(\underline{Y})$, with respect to the matrix $\underline{B}$ is

$$\frac{\partial J(\underline{Y})}{\partial \underline{B}} = -2 \begin{bmatrix} \sinh y_{11} & \sinh y_{12} & \cdots & \sinh y_{1,R-1} \\ \cdot & \cdot & \cdots & \cdot \\ \sinh y_{N1} & \sinh y_{N2} & \cdots & \text{siny } y_{N,R-1} \end{bmatrix}$$

$$= -2\ \underline{S}(\underline{Y}) \tag{5.38}$$

Since $\underline{U}$ is not constrained in any manner, $\frac{\partial J(\underline{Y})}{\partial \underline{U}} = \underline{0}$ implies that $\underline{S}(\underline{Y}) = \underline{0}$, which, in turn, implies that $\sinh y_{ij} = 0$ and hence $y_{ij} = 0$ for all $i=1,\ldots,N$ and $j=1,2,\ldots,R-1$. Therefore, for $\frac{\partial J(\underline{Y})}{\partial \underline{U}} = \underline{0}$ and a fixed $\underline{B}$,

$$\underline{Y} = \underline{A}\ \underline{U} - \underline{B} = \underline{0}$$

which gives a least square fit of

$$\underline{U} = \underline{A}^{\#}\underline{B}\ . \tag{5.39}$$

On the other hand, for a fixed $\underline{U}$ and the constraint $\underline{B}_j(\underline{e}_j - \underline{e}_i) > 0$ for all $i \neq j$ as given in (5.13), $\underline{B}$ may be incremented according to the following gradient descent procedure to reduce $J(\underline{Y})$ at each step,

$$\underline{B}(k+1) = \underline{B}(k) + \delta\underline{B}(k) \tag{5.40}$$

where the q-th element, $\delta[{}_{\ell}b_{jq}(k)]$, of $\delta[{}_{\ell}\underline{b}_j^{\,t}(k)]$ in $\delta\underline{B}_j(k)$ is given by

$$\delta[{}_{\ell}b_{jq}(k)] = \begin{cases} -p(k)\ {}_{\ell}\left[\dfrac{\partial J(\underline{Y})(k))}{\partial \underline{B}}\right]_{jq} = 2p(k)\ {}_{\ell}\underline{S}_{jq}(\underline{Y}(k)), \\ \qquad\qquad \text{if } {}_{\ell}\underline{Y}_j(k)(\underline{e}_j-\underline{e}_q) > 0 \text{ for any } q\neq j \\ 0 \qquad\qquad \text{if } {}_{\ell}\underline{Y}_j(k)(\underline{e}_j-\underline{e}_q) \leq 0 \text{ for any } q\neq j\ . \end{cases}$$

However, ${}_{\ell}\underline{Y}_j(k)(\underline{e}_j-\underline{e}_q) > 0$ does not imply ${}_{\ell}\underline{S}_j(\underline{Y}(k))(\underline{e}_j-\underline{e}_q) > 0$. In order to make $\delta[{}_{\ell}\underline{b}_j^{\,t}(k)](\underline{e}_j-\underline{e}_q) \geq 0$ so that (5.13) can be satisfied at each step, a modified gradient descent procedure is to be used. Let a (R-1) by (R-1) non-singular matrix $\underline{E}_j$ be defined as

$$\underline{E}_j = [\underline{e}_j - \underline{e}_1, \ldots, \underline{e}_j - \underline{e}_{j-1}, \underline{e}_j - \underline{e}_{j+1}, \ldots, \underline{e}_j - \underline{e}_R]. \qquad (5.41)$$

Also define

$$\underline{Z}_j = \underline{Y}_j\,\underline{E}_j \qquad \text{for all } j=1,2,\ldots,R. \qquad (5.42)$$

The increment $\delta[{}_{\ell}b_{jq}(k)]$ is then given in terms of

$$\delta[{}_{\ell}\underline{b}_j^{\,t}(k)\underline{E}_j]_q = \begin{cases} 2\,p(k)\ {}_{\ell}S_{jq}(\underline{Z}(k)) = p(k)[{}_{\ell}S_{jq}(\underline{Z}(k)) + {}_{\ell}\Lambda_{jq}(k)] \\ \qquad\qquad \text{if } {}_{\ell}Z_{jq}(k) = {}_{\ell}\underline{Y}_j(k)(\underline{e}_j-\underline{e}_q) > 0 \\ 0 \qquad\qquad \text{if } {}_{\ell}Z_{jq}(k) = {}_{\ell}\underline{Y}_j(k)(\underline{e}_j-\underline{e}_q) \leq 0 \end{cases}$$

(5.43)

where

$$ {}_{\ell}\Lambda_{jq}(k) = {}_{\ell}S_{jq}(\underline{Z}(k))\ \mathrm{Sgn}\ ({}_{\ell}Z_{jq}(k)) \tag{5.44}$$

and, following (5.33),

$$ {}_{\ell}S_{jq}(\underline{Z}(k)) = \mathrm{Sinh}\ {}_{\ell}Z_{jq}(k). \tag{5.45}$$

Putting into vector representation,

$$\delta[{}_{\ell}\underline{b}_j(k)\underline{E}_j] = p(k)\ [{}_{\ell}\underline{S}_j(\underline{Z}(k)) + {}_{\ell}\underline{\Lambda}_j(k)]$$

or

$$\begin{aligned}\delta[{}_{\ell}\underline{b}_j(k)] &= p(k)\ [{}_{\ell}\underline{S}_j(\underline{Z}(k)) + {}_{\ell}\underline{\Lambda}_j(k)]\underline{E}_j^{-1} \\ &= p(k)\ {}_{\ell}\underline{H}_j(\underline{Y}(k))\end{aligned} \tag{5.46}$$

where

$$ {}_{\ell}\underline{H}_j(\underline{Y}(k)) = [{}_{\ell}\underline{S}_j(\underline{Z}(k)) + {}_{\ell}\underline{\Lambda}_j(k)]\underline{E}_j^{-1}\ . \tag{5.47}$$

$$\underline{H}_j(\underline{Y}(k)) = [\underline{S}_j(\underline{Z}(k)) + \underline{\Lambda}_j(k)]\underline{E}_j^{-1}$$

$$\underline{H}(\underline{Y}(k)) = \begin{bmatrix} \underline{H}_1(\underline{Y}(k) \\ \vdots \\ \underline{H}_j(\underline{Y}(k)) \\ \vdots \\ \underline{H}_R(\underline{Y}(k)) \end{bmatrix} = \begin{bmatrix} {}_1\underline{H}_1(\underline{Y}(k)) \\ \vdots \\ {}_{\ell}\underline{H}_j(\underline{Y}(k)) \\ \vdots \\ {}_{n_R}\underline{H}_R(\underline{Y}(k)) \end{bmatrix}$$

$$= [\underline{h}_1(\underline{Y}(k))\ \ldots\ \underline{h}_q(\underline{Y}(k))\ \ldots\ \underline{h}_{R-1}(\underline{Y}(k))]. \tag{5.48}$$

It follows from (5.46) and (5.44) that

$$\delta[{}_{\ell}\underline{b}_j(k)](\underline{e}_j - \underline{e}_i) \geq 0 \text{ for all } i \neq j \text{ and for all } j.$$

Then

$$\delta[\underline{B}(k)] = p(k)\ \underline{H}(\underline{Y}(k))$$

Substituting the above equation into (5.40), one has

$$\underline{B}(k+1) = \underline{B}(k) + p(k)\ \underline{H}(\underline{Y}(k)) \tag{5.49}$$

Using the above equation in (5.39), one has

$$\begin{aligned} \underline{U}(k+1) &= \underline{A}^{\#}\underline{B}(k+1) = \underline{A}^{\#}\ \{\underline{B}(k) + p(k)\ \underline{H}[\underline{Y}(k)]\} \\ &= \underline{U}(k) + p(k)\ \underline{A}^{\#}\underline{H}[\underline{Y}(k)] \end{aligned} \tag{5.50}$$

Therefore, an iterative algorithm to solve for $\underline{U}$ can be proposed in the following:

$$\left[\begin{array}{l} \underline{U}(0) = \underline{A}^{\#}\underline{B}(0) \\ \underline{Y}(k) = \underline{A}\ \underline{U}(k) - \underline{B}(k),\ \underline{Z}_j(k) = \underline{Y}_j(k)\ \underline{E}_j \\ \underline{B}(k+1) = \underline{B}(k) + p(k)\ \underline{H}[\underline{Y}(k)],\ \underline{H}_j(\underline{Y}(k)) = [\underline{S}_j(k)) + \underline{\Lambda}_j(k)]\underline{E}_j^{-1} \\ \underline{U}(k+1) = \underline{U}(k) + p(k)\ \underline{A}^{\#}\ \underline{H}[\underline{Y}(k)] \end{array}\right. \tag{5.51}$$

where p(k) may be chosen as equal to

$$p(k) = \frac{\sum_{j=1}^{R} \sum_{\ell=1}^{n_j} \{{}_{\ell}\varepsilon_j(k) + {}_{\ell}\underline{H}_j(\underline{Y}(k))(\underline{E}_j^{\,t})^{-1}\underline{R}(\ \underline{Z}_j(k))\underline{E}_j^{\,t}\ {}_{\ell}\underline{H}_j(\underline{Y}(k))\}}{2\sum_{q=1}^{R-1} \underline{h}_q^{\,t}\ (\underline{I} - \underline{A}\ \underline{A}^{\#})\ \underline{h}_q} \tag{5.52}$$

provided that

$$\sum_{j=1}^{R} \sum_{\ell=1}^{n_j} \{{}_{\ell}\varepsilon_j(k) + {}_{\ell}\underline{H}_j(\underline{Y}(k))(\underline{E}_j^{\,t})\underline{R}({}_{\ell}\underline{Z}_j(k))\underline{E}_j^{\,t}\ {}_{\ell}\underline{H}_j(\underline{Y}(k))\} > 0 \tag{5.53}$$

where ${}_{\ell}\varepsilon_j(k)$ and $\underline{R}({}_{\ell}\underline{Z}_j(k))$ are defined in (5.62) and (5.60) respectively as will be shown later. The initial $\underline{B}$ matrix, $\underline{B}(0)$, may be chosen from

$$\underline{B}(0) = \beta \begin{bmatrix} \underline{e}^t{}_1 \\ \vdots \\ \underline{e}^t{}_1 \\ \hline \vdots \\ \hline \underline{e}^t{}_j \\ \vdots \\ \underline{e}^t{}_j \\ \hline \vdots \\ \hline \underline{e}^t{}_R \\ \vdots \\ \underline{e}^t{}_R \end{bmatrix}, \quad \beta > 0 \tag{5.54}$$

where $\underline{e}_j$'s are vertex vectors of (R-1)-dimensional simplex and β is an arbitrary positive constant. A recursive relation in $\underline{Y}(k)$ is also obtained as follows:

$$\underline{Y}(k+1) = \underline{Y}(k) + p(k)(\underline{A}\ \underline{A}^{\#} - \underline{I})\underline{H}[\underline{Y}(k)] \tag{5.55}$$

Compare (5.51) and (2.29), it is evident that the above algorithm for multiclass pattern classification is a generalization of the dichotomy algorithm developed in Chapter II.

C. Theorem 2

In order to prove the convergence of the algorithm (5.51), the following discussion is necessary.

<u>Lemma 2</u>. Consider the set of inequalities (5.19) and the algorithm (5.51) to solve it. Then

1) $\underline{Y}_j(k)(\underline{e}_j-\underline{e}_i) \ngeq \underline{0}$ for all $i \neq j$

for all $j=1,2,\ldots,R$

for any k

2) If (5.19) is consistent, then

$\underline{Y}_j(k)(\underline{e}_j-\underline{e}_i) \nleq \underline{0}$ for all $i \neq j$

for all $j=1,2,\ldots,R$

for any k

<u>Proof</u>.

1) Let

$$\underline{Y}_j(k)(\underline{e}_j-\underline{e}_i) \geq \underline{0} \text{ for all } i \neq j$$

for all $j=1,2,\ldots,R$

for some k

Since

$$\underline{B}_j(k)(\underline{e}_j-\underline{e}_i) > \underline{0} \text{ for all } i \neq j$$

Then

$$(\underline{e}_j-\underline{e}_i)^t \underline{Y}_j^t(k)\underline{B}_j(k)(\underline{e}_j-\underline{e}_i) > 0 \text{ for all } i \neq j$$

$$\underline{Y}_j^t(k)\ \underline{B}_j(k) > 0 \qquad \text{for all } j=1,2,\ldots,R$$

it follows that

$$\underline{Y}(k)\underline{B}(k) > 0$$

But

$$\underline{Y}(k) = (\underline{A}\ \underline{A}^{\#} - \underline{I})\ \underline{B}(k),$$

$$\underline{Y}^{t}(k)\underline{B}(k) = \underline{B}^{t}(k)\ (\underline{A}\ \underline{A}^{\#} - \underline{I})\ \underline{B}(k) \leq 0$$

since

$$(\underline{A}\ \underline{A}^{\#} - \underline{I}) \leq 0 \ .$$

This is a contradition. Hence

$$\underline{Y}_j(k)\ (\underline{e}_j - \underline{e}_i) \nleq 0 \quad \text{for all } i \neq j$$
$$\text{for all } j = 1,2,\ldots,R$$
$$\text{for any } k.$$

2) Assume that (5.19) is consistent but

$$\underline{Y}_j(k)\ (\underline{e}_j - \underline{e}_i) \leq \underline{0} \quad \text{for all } i \neq j$$
$$\text{for all } j = 1,2,\ldots,R$$
$$\text{for some } k$$

Consistence of (5.19) implies the existence of a $\underline{U}^{*}$ and a $\underline{B}^{*}$ such that

$$\underline{A}\ \underline{U}^{*} = \underline{B}^{*}$$

and

$$\underline{A}_j\underline{U}^* \ (\underline{e}_j-\underline{e}_i) = \underline{B}_j^*(\underline{e}_j-\underline{e}_i) > \underline{0} \text{ for all } i\neq j,$$
$$\text{for all } j=1,2,\ldots,R$$

Therefore

$$(\underline{e}_j-\underline{e}_i)^t \ \underline{Y}^t{}_j(k) \ \underline{B}_j^*(\underline{e}_j-\underline{e}_i) < 0 \qquad \text{for all } i\neq j$$
$$\underline{Y}^t{}_j(k) \ \underline{B}^*{}_j < 0 \qquad \text{for all } j=1,2,\ldots,R$$

and

$$\underline{Y}^t(k) \ \underline{B}^* = \sum_{j=1}^{R} \underline{Y}^t{}_j(k) \ \underline{B}^*{}_j < 0.$$

But for any $\underline{Y}(k)$,

$$\underline{A}^t\underline{Y}(k) = \underline{A}^t(\underline{A} \ U(k) - B(k)) = \underline{A}^t(\underline{A} \ \underline{A}^\# - \underline{I}) \ \underline{B}(k)$$
$$= (\underline{A}^t\underline{A} \ \underline{A}^\# - \underline{A}^t) \ \underline{B}(k) = \underline{0} \ \underline{B}(k) = \underline{0} \ .$$

thus

$$\underline{U}^{*t} \ \underline{A}^t \ \underline{Y}(k) = 0.$$

or

$$\underline{Y}^t(k) \ \underline{A} \ \underline{U}^* = \underline{Y}^t(k) \ \underline{B}^* = \sum_{j=1}^{R} \underline{Y}^t{}_j(k) \ \underline{B}^*{}_j = \underline{0}$$

which is a contradition. Hence, if (5.19) is consistent,

$$\underline{Y}_j(k) \ (\underline{e}_j-\underline{e}_i) \not\leq \underline{0} \qquad \begin{array}{l}\text{for all } i\neq j \\ \text{for all } j=1,2,\ldots,R \\ \text{for any } k.\end{array}$$

<u>Theorem 2</u>. Consider the set of linear inequalities (5.19) and the algorithm (5.51) to solve them, and let

$$V[\underline{Y}(k)] = ||\underline{Y}(k)|| \triangleq Tr[\underline{Y}^t(k)\underline{Y}(k)] = \sum_{q=1}^{R-1} ||\underline{y}_q(k)||^2 = \sum_{j=1}^{R} \sum_{i=1}^{n_j} ||{}_i\underline{Y}_j(k)||^2$$

1) If the set of linear inequalities is consistent, then

a) $\Delta V[\underline{Y}(k)] \triangleq V[\underline{Y}(k+1)] - V[\underline{Y}(k)] < 0$ and

$\lim_{k\to\infty} V[\underline{Y}(k)] = 0$ implying convergence to a solution in an infinite number of iterations; and

b) Actually, a solution is obtained in a finite number of steps.

2) If the set of linear inequalities is inconsistent, then there exists a positive integer k^* such that

$$\Delta V[\underline{Y}(k)] < 0 \qquad \text{for } k < k^*$$

$$\Delta V[\underline{Y}(k)] = 0 \qquad \text{for } k \geq k^*$$

$${}_\ell\underline{Y}_j(k)(\underline{e}_j - \underline{e}_i) \nleq 0 \qquad \begin{array}{l}\text{for } k < k^* \\ \text{for all } i \neq j \\ \text{for all } j=1,2,\ldots,R\end{array}$$

$${}_\ell\underline{Y}_j(k)(\underline{e}_j - \underline{e}_i) = {}_i\underline{Y}_j(k^*)(\underline{e}_j - \underline{e}_i) \leq 0 \qquad \begin{array}{l}\text{for all } k \geq k^* \\ \text{for all } i \neq j \\ \text{for all } j=1,2,\ldots,R\end{array}$$

and

$$\underline{U}(k) = \underline{U}(k^*) \qquad \text{for } k \geq k^*$$

$$\underline{B}(k) = \underline{B}(k^*) \qquad \text{for } k \geq k^*$$

In other words, the occurrence of a matrix $\underline{Y}(k)$ with all non-positive elements of $\underline{Y}(k)\cdot(\underline{e}_j-\underline{e}_i)$ for all $i\neq j$ and all j at any step terminates the algorithm and indicates the nonlinear separability of the R-class patterns.

Proof.

The proof of this theorem is similar to the convergence proof of the generalization of Ho-Kashyap algorithm to multiclass pattern classification[23].

Part 1:

With reference to the recursive relation in $\underline{Y}(k)$ given by (5.55), $V[\underline{Y}(k)]$ can be considered as a Liapunov function,

$$V[\underline{Y}(k)] = Tr[\underline{Y}^t(k)\underline{Y}(k)] > 0 \text{ for all } \underline{Y}(k)\neq 0 \ . \qquad (5.56)$$

Now,

$$\begin{aligned}\Delta V[\underline{Y}(k)] &\triangleq V[\underline{Y}(k+1)] - V[\underline{Y}(k)]\\ &= Tr[\underline{Y}^t(k+1)\underline{Y}(k+1) - \underline{Y}^t(k)\underline{Y}(k)]. \qquad (5.57)\end{aligned}$$

Since

$$\begin{aligned}&\underline{Y}^t(k+1)\underline{Y}(k+1) - \underline{Y}^t(k)\ \underline{Y}(k)\\ &= [\underline{Y}^t(k) + p(k)\underline{H}^t(\underline{Y}(k))(\underline{A}\ \underline{A}^{\#}-\underline{I})][\underline{Y}(k)+p(k)(\underline{A}\ \underline{A}^{\#}-\underline{I})H(\underline{Y}(k))]-\underline{Y}^t(k)\underline{Y}(k)\\ &= p(k)\ \underline{H}^t[\underline{Y}(k)](\underline{A}\ \underline{A}^{\#}-\underline{I})\underline{Y}(k) + p(k)\underline{Y}^t(k)(\underline{A}\ \underline{A}^{\#}-\underline{I})\underline{H}[\underline{Y}(k)]\\ &\quad +p^2(k)\ \underline{H}^t[\underline{Y}(k)]\ (\underline{I} - \underline{A}\ \underline{A}^{\#})\underline{H}[\underline{Y}(k)],\end{aligned}$$

and

$$\underline{A}\ \underline{A}^{\#}\underline{Y}(k) = \underline{A}\ \underline{A}^{\#}[\underline{A}\ \underline{U}(k) - \underline{B}(k)] = \underline{A}\ \underline{A}^{\#}[\underline{A}\ \underline{A}^{\#}\underline{B}(k) - \underline{B}(k)]$$

$$= \underline{A}\ \underline{A}^{\#}\underline{B}(k) - \underline{A}\ \underline{A}^{\#}\underline{B}(k) = \underline{0}.$$

then

$$V[\underline{Y}(k)] = Tr\{-2p(k)\underline{H}^{t}[\underline{Y}(k)]\underline{Y}(k)+p^{2}(k)\underline{H}^{t}[\underline{Y}(k)](\underline{I}-\underline{A}\ \underline{A}^{\#})\underline{H}[\underline{Y}(k)]\}$$

$$= -2p(k)\ Tr\ \{\underline{H}^{t}[\underline{Y}(k)]\underline{Y}(k)\}+ p^{2}(k)Tr\{\underline{H}^{t}[\underline{Y}(k)](\underline{I}-\underline{A}\ \underline{A}^{\#})\underline{H}[\underline{Y}(k)]\}$$

$$= -2p(k)Tr\{\underline{H}[\underline{Y}(k)\underline{Y}^{t}(k)\} + p^{2}(k)\sum_{q=1}^{R-1}\underline{h}^{t}{}_{q}[\underline{Y}(k)](\underline{I}-\underline{A}\ \underline{A}^{\#})\underline{h}_{q}[\underline{Y}(k)]$$

$$= -2p(k)\sum_{j=1}^{R}\sum_{\ell=1}^{n_j}{}_{\ell}\underline{H}_{j}(\underline{Y}(k)_{\ell}\underline{Y}_{j}{}^{t}(k)+p^{2}(k)\sum_{q=1}^{R-1}\underline{h}_{q}{}^{t}(\underline{Y}(k))(\underline{I}-\underline{AA}^{\#})\underline{h}_{q}(\underline{Y}$$

(5.58)

From (5.45) and (5.33),

$${}_{\ell}\underline{S}_{j}(\underline{Z}) = [\frac{\sinh\ {}_{\ell}Z_{j1}}{{}_{\ell}Z_{j1}}\ {}_{\ell}Z_{j1},\ \ldots,\ \frac{\sinh\ {}_{\ell}Z_{j,R-1}}{{}_{\ell}Z_{j,R-1}}\ {}_{\ell}Z_{j,R-1}]$$

$$= {}_{\ell}\underline{Z}_{j}\begin{bmatrix}\frac{\sinh\ {}_{\ell}Z_{j1}}{{}_{\ell}Z_{j1}} & 0 & \cdots\cdots & 0 \\ \hline 0 & & S & \frac{\sinh\ {}_{\ell}Z_{j,R-1}}{{}_{\ell}Z_{j,R-1}}\end{bmatrix}$$

$$= {}_{\ell}\underline{Z}_{j}\underline{\mathbf{R}}(\ \underline{Z}_{j}) \qquad (5.59)$$

where

$$\underline{R}({}_{\ell}\underline{Z}_j) \stackrel{\Delta}{=} \text{diag}\ [r_{11}({}_{\ell}\underline{Z}_j),\ \ldots,\ r_{R-1,R-1}({}_{\ell}\underline{Z}_j)] \tag{5.60}$$

$$(j=1,2,\ldots,R;\quad \ell=1,2,\ldots,n_j)$$

$$r_{qq}({}_{\ell}\underline{Z}_j) \stackrel{\Delta}{=} \frac{\text{Sinh}\ {}_{\ell}Z_{jq}}{{}_{\ell}Z_{jq}} \geq 1,\ (q=1,\ldots,R-1).$$

Substituting (5.59) into (5.47),

$${}_{\ell}\underline{H}_j(\underline{Y}(k)) = [{}_{\ell}\underline{Z}_j(k)\underline{R}({}_{\ell}\underline{Z}_j(k) + {}_{\ell}\underline{\Lambda}_j(k)]\underline{E}_j^{-1}$$

then

$$-2p \sum_{j=1}^{R} \sum_{\ell=1}^{n_j} {}_{\ell}\underline{H}_j\ {}_{\ell}\underline{Y}_j^{\,t}$$

$$= -2p \sum_j \sum_\ell [{}_{\ell}\underline{Z}_j\underline{R}({}_{\ell}\underline{Z}_j) + {}_{\ell}\underline{\Lambda}_j]\ \underline{E}_j^{-1}(\underline{E}_j^{\,t})^{-1}\ {}_{\ell}\underline{Z}_j^{\,t}$$

$$= -2p \sum_j \sum_\ell [{}_{\ell}\underline{Z}_j\underline{R}({}_{\ell}\underline{Z}_j)+{}_{\ell}\underline{\Lambda}_j]\underline{E}_j^{-1}(\underline{E}_j^{\,t})^{-1}\underline{R}^{-1}({}_{\ell}\underline{Z}_j)[{}_{\ell}\underline{Z}_j\underline{R}({}_{\ell}\underline{Z}_j)]^t$$

$$= -p \sum_j \sum_\ell [{}_{\ell}\underline{Z}_j\underline{R}({}_{\ell}\underline{Z}_j) + {}_{\ell}\underline{\Lambda}_j]\underline{E}_j^{-1}(\underline{E}_j^{\,t})^{-1}\underline{R}^{-1}({}_{\ell}\underline{Z}_j)[{}_{\ell}\underline{Z}_j\underline{R}({}_{\ell}\underline{Z}_j) + {}_{\ell}\underline{\Lambda}_j]^t$$

$$-p \sum_j \sum_\ell [{}_{\ell}\underline{Z}_j\underline{R}({}_{\ell}\underline{Z}_j) + {}_{\ell}\underline{\Lambda}_j]\underline{E}_j^{-1}(\underline{E}_j^{\,t})^{-1}\underline{R}^{-1}({}_{\ell}\underline{Z}_j)[{}_{\ell}\underline{Z}_j\underline{R}({}_{\ell}\underline{Z}_j) - {}_{\ell}\underline{\Lambda}_j]^t$$

$$= -p \sum_j \sum_\ell {}_{\ell}\underline{H}_j(\underline{Y}(k))(\underline{E}_j^{\,t})^{-1}\ \underline{R}^{-1}({}_{\ell}\underline{Z}_j)\underline{E}_j^{\,t}\,{}_{\ell}\underline{H}_j^{\,t}(\underline{Y}(k))$$

$$-p \sum_j \sum_\ell [{}_{\ell}\underline{Z}_j\underline{R}({}_{\ell}\underline{Z}_j) + {}_{\ell}\underline{\Lambda}_j](\underline{E}_j^{\,t}\underline{E}_j)^{-1}\underline{R}^{-1}({}_{\ell}\underline{Z}_j\underline{R}({}_{\ell}\underline{Z}_j) - {}_{\ell}\underline{\Lambda}_j]^t. \tag{5.61}$$

Since the off diagonal elements in $(\underline{E}_j^{\,t}\underline{E}_j)^{-1}$ are negative[(23)] and $\underline{R}^{-1}({}_{\ell}\underline{Z}_j)$ is a diagonal matrix with all positive diagonal elements, the off diagonal elements of $(\underline{E}_j^{\,t}\underline{E}_j)^{-1}\ \underline{R}^{-1}({}_{\ell}\underline{Z}_j)$ are also negative. From (5.44), (5.47), and (5.49), the elements of $({}_{\ell}\underline{Z}_j\underline{R}(\ \underline{Z}_j) + {}_{\ell}\underline{\Lambda}_j]$ are either positive or zero, and

the corresponding elements of $[{}_{\ell}\underline{Z}_j\underline{R}({}_{\ell}\underline{Z}_j) + {}_{\ell}\underline{\Lambda}_j]$ are either zero or negative. Hence,

$$
{}_{\ell}\varepsilon_j \overset{\Delta}{=} [{}_{\ell}\underline{Z}_j\underline{R}({}_{\ell}\underline{Z}_j) + {}_{\ell}\underline{\Lambda}_j]\ (\underline{E}_j{}^t\underline{E}_j)^{-1}\ \underline{R}^{-1}({}_{\ell}\underline{Z}_j)[{}_{\ell}\underline{Z}_j\underline{R}({}_{\ell}\underline{Z}_j) - {}_{\ell}\underline{\Lambda}_j]^t \geq 0
$$

for all j and all ℓ. (5.62)

Substituting (5.62) into (5.61) which in turn, is substituted into (5.58), one obtains

$$
\begin{aligned}
\Delta V[\underline{Y}(k)] = & -p(k) \sum_j \sum_{\ell} {}_{\ell}\underline{H}_j(\underline{Y}(k))(\underline{E}_j{}^t)^{-1}\underline{R}^{-1}({}_{\ell}\underline{Z}_j)\underline{E}_j{}^t{}_{\ell}\underline{H}_j(\underline{Y}(k)) \\
& -p(k) \sum_j \sum_{\ell} {}_{\ell}\varepsilon_j(k) + p^2(k) \sum_q \underline{h}_q{}^t(\underline{Y}(k))(\underline{I} - \underline{A}\ \underline{A}^{\#})\underline{h}_q(\underline{Y}(k)) \\
= & -p(k) \sum_j \sum_{\ell} {}_{\ell}\varepsilon_j(k) + {}_{\ell}\underline{H}_j(\underline{Y}(k))(\underline{E}_j{}^t)^{-1}\underline{R}^{-1}({}_{\ell}\underline{Z}_j)\underline{E}_j{}^t{}_{\ell}\underline{H}_j{}^t(\underline{Y}(k)) \\
& + p^2(k) \sum_q \underline{h}_q{}^t(\underline{Y}(k))(\underline{I} - \underline{A}\ \underline{A}^{\#})\underline{h}_q(\underline{Y}(k)) \\
= & -p(k) \sum_{j=1}^{R} \sum_{\ell=1}^{n_j} {}_{\ell}\varepsilon_j(k) - p^2(k) \sum_{q=1}^{R-1} \underline{h}_q{}^t(\underline{Y}(k))\underline{A}\ \underline{A}^{\#}\underline{h}_q(\underline{Y}(k)) \\
& - p(k) \sum_{j=1}^{R} \sum_{\ell=1}^{n_j} {}_{\ell}\underline{H}_j(\underline{Y}(k))\ \{(\underline{E}_j{}^t)^{-1}[\underline{R}^{-1}({}_{\ell}\underline{Z}_j) - p(k)\underline{I}]\underline{E}_j{}^t\}{}_{\ell}\underline{H}_j{}^t(\underline{Y}(k)).
\end{aligned}
$$

(5.63)

$V(\underline{Y}(k))$ is negative definite if the right hand side of the above equation is negative definite in $[{}_{\ell}\underline{Z}_j\underline{R}({}_{\ell}\underline{Z}_j) + {}_{\ell}\underline{\Lambda}_j]$. The first two terms on the right hand side are negative semi-definite. If a value of p(k) can be found such that

$$\sum_{j=1}^{R} \sum_{\ell=1}^{n_j} {}_{\ell}\underline{H}_j(\underline{Y}(k)) \{(\underline{E}_j^{\,t})^{-1}[\underline{R}^{-1}({}_{\ell}\underline{Z}_j) - p(k)\underline{I}]\underline{E}_j^{\,t}\} \, {}_{\ell}\underline{H}_j(\underline{Y}(k)) > 0$$

then $\Delta V(\underline{Y}(k))$ is negative definite in $[{}_{\ell}\underline{Z}_j \underline{R}({}_{\ell}\underline{Z}_j) + {}_{\ell}\underline{\Lambda}_j]$. Note that when

$$p(k) = \frac{1}{\cosh Y_{max}(k)} , \qquad Y_{max}(k) = \underset{j,\ell,q}{\mathrm{Max}} \; |{}_{\ell}Y_{jq}(k)| ,$$

$[\underline{R}^{-1}({}_{\ell}\underline{Z}_j) - p(k)\underline{I}]$ is positive definite and has real eigenvalues as can be shown by following (2.43) and (2.44); but it is not certain that $(\underline{E}_j^{\,t})^{-1}\,[\underline{R}^{-1}({}_{\ell}\underline{Z}_j) - p(k)\underline{I}]\,\underline{E}_j^{\,t}$ can be positive definite for all j and all ℓ. Let p(k) be so chosen as to maximize $-\Delta V[\underline{Y}(k)]$ at each step, one follows the procedure used in Section II-C to obtain a choice of p(k) as given in (5.52), provided (5.53) is satisfied to make sure that p(k) > 0. For this value of p(k),

$$\Delta V(\underline{Y}(k)) = - \frac{\left[\sum_{j=1}^{R} \sum_{\ell=1}^{n_j} \{{}_{\ell}\varepsilon_j(k) + {}_{\ell}\underline{H}_j(\underline{Y}(k))(\underline{E}_j^{\,t})^{-1}\,\underline{R}({}_{\ell}\underline{Z}_j(k))\underline{E}_j^{\,t}\,{}_{\ell}\underline{H}_j(\underline{Y}(k))\}\right]^2}{4 \sum_{q=1}^{R-1} \underline{h}_q^{\,t}(\underline{I} - \underline{A}\,\underline{A}^{\#})\,\underline{h}_q}$$

$$\leq 0 \quad \text{for } [{}_{\ell}\underline{Z}_j \underline{R}({}_{\ell}\underline{Z}_j) + {}_{\ell}\underline{\Lambda}_j] \neq 0 .$$

Hence, $V[\underline{Y}(k)]$ is negative definite in $[{}_{\ell}\underline{Z}_j\ \underline{R}({}_{\ell}\underline{Z}_j) + {}_{\ell}\underline{\Lambda}_j]$. Note that ${}_{\ell}\underline{Z}_j\ \underline{R}({}_{\ell}\underline{Z}_j) + {}_{\ell}\underline{\Lambda}_j = \underline{0}$ for all j and all ℓ only if ${}_{\ell}\underline{Z}_j \leq 0$, that is only if $\underline{Y}(k) = \underline{0}$ or ${}_{i}\underline{Y}_j(k)(\underline{e}_j - \underline{e}_i) \leq 0$ for all $i \neq j$ and for all j. Since it is assumed that the set of the inequalities (5.19) is consistent, from the lemma $\underline{Y}_j(k)(\underline{e}_j - \underline{e}_i) \nleq 0$ for all $i \neq j$ and for all j, therefore

$$\Delta V[\underline{Y}(k)] < 0 \text{ for all } \underline{Y}(k) \neq \underline{0}$$

$$= 0 \text{ if } \underline{Y}(k) = \underline{0} \qquad (5.64)$$

and the solution $\underline{Y} = \underline{0}$ of equation (5.55) can be reached asymptotically, that is

$$\lim_{k \to \infty} ||\underline{Y}(k)||^2 = 0$$

which corresponds to a solution $\underline{U}^{**}$ with $\underline{A}\ \underline{U}^{**} = \underline{B}$ such that $\underline{A}_j\underline{U}^{**}(\underline{e}_j - \underline{e}_i) = \underline{B}_j(\underline{e}_j - \underline{e}_i) > \underline{0}$ for all $i \neq j$ and for all j. This completes the proof of Part 1(a).

Note that if the $\underline{B}(0)$ given in (5.54) is

$${}_{\ell}\underline{b}^t{}_j(0)(\underline{e}_j - \underline{e}_i) = \underline{e}^t{}_j(\underline{e}_j - \underline{e}_i) > 0 \text{ for all } i \neq j \text{ for all } j \qquad (5.65)$$

Then the algorithm (5.51) gives

$$ {}_{\ell}\underline{b}_j^{\,t}(k+1) = {}_{\ell}\underline{b}^{t}_{\,j}(k) + p(k)[\,{}_{\ell}\underline{S}_j(\underline{Z}(k)) + {}_{\ell}\underline{\Delta}_j(k)]\underline{E}_j^{-1} $$

$$ (j=1,2,\ldots,R,\quad \ell=1,2,\ldots,n_j) \tag{5.66} $$

which implies

$$ {}_{\ell}\underline{b}^{t}_{\,j}(k+1)(\underline{e}_j-\underline{e}_i) > {}_{\ell}\underline{b}^{t}_{\,j}(0)(\underline{e}_j-\underline{e}_i). \tag{5.67} $$

From (5.65), (5.66) and (5.67), by induction

$$ {}_{i}\underline{b}^{t}_{\,j}(k+1)(\underline{e}_j-\underline{e}_i) > (1+\varepsilon)\underline{e}^{t}_{\,j}(\underline{e}_j-\underline{e}_i),\ \varepsilon > 0. $$

$$ \begin{array}{l} \text{for all } i\neq j \\ \text{for all } j \\ \text{for all } k \end{array} \tag{5.68} $$

which satisfies the condition given in (5.13). Since

$$ V[\underline{Y}(k)] = ||\underline{Y}(k)|| = \sum_{j=1}^{R}\ \sum_{i=1}^{n_j}\ ||{}_{i}\underline{Y}_j(k)||^2 < 1 $$

at a certain finite k, it implies that

$$ ||{}_{i}\underline{Y}_j(k)||^2 < 1 $$

and

$$ {}_i\underline{Y}_j(k)(\underline{e}_j-\underline{e}_i) > -\underline{e}^t{}_j(\underline{e}_j-\underline{e}_i) \text{ if } k \text{ is sufficiently large.} \qquad (5.69)$$

$$\text{for all } i \neq j$$
$$\text{for all } j$$

Let $\underline{v}_j$ be an n by 1 vector whose components all equal unity,

$$\underline{v}^t{}_j = [1,1,1,\ldots,1] \qquad (5.70)$$

From (5.68) and (5.69),

$$\underline{B}_j(k)(\underline{e}_j-\underline{e}_i) > (1+\varepsilon)\underline{e}^t{}_j(\underline{e}_j-\underline{e}_i)\underline{v} \text{ for all } i\neq j \qquad (5.71)$$

$$\underline{Y}_j(k)(\underline{e}_j-\underline{e}_i) \quad -\underline{e}^t{}_j(\underline{e}_j-\underline{e}_i)\underline{v} \text{ for all } i\neq j \qquad (5.72)$$

Since

$$\underline{A}_j\underline{U}(k) = \underline{B}_j(k) + \underline{Y}_j(k)$$

it follows that

$$\begin{aligned}
\underline{A}_j\underline{U}(k)(\underline{e}_j-\underline{e}_i) &= \underline{B}_j(k)(\underline{e}_j-\underline{e}_i) + \underline{Y}_j(k)(\underline{e}_j-\underline{e}_i) \\
&> (1+\varepsilon)\underline{e}^t{}_j(\underline{e}_j-\underline{e}_i)\underline{v} - \underline{e}^t{}_j(\underline{e}_j-\underline{e}_i)\underline{v} \\
&> \varepsilon\, \underline{e}^t{}_j(\underline{e}_j - \underline{e}_i)\underline{v} \\
&> 0 \text{ for all } i\neq j \\
&\quad \text{for all } j=1,2,\ldots,R \qquad (5.73)
\end{aligned}$$

which indicates a solution $\underline{U}^* = \underline{U}(k)$ is obtained in a finite number of steps. This completes the proof of Part 1(b).

Part 2:

If the set of inequalities (5.19) is inconsistent, $\underline{Y}(k)$ cannot be $\underline{0}$ and hence $V[\underline{Y}(k)]$ cannot become zero for any $k > 0$. There must exist a value of k, $k=k^*$, such that

$$\Delta V[\underline{Y}(k)] < 0 \text{ for } 0 \leq k < k^*$$
$$= 0 \text{ for } k = k^*$$

But as shown in Part 1, $\Delta V[\underline{Y}(k^*)] = 0$ only if either $\underline{Y}(k^*) = \underline{0}$ or ${}_i\underline{Y}_j(k^*)(\underline{e}_j - \underline{e}_i) \leq 0$ for all $i \neq j$ and for all j. Since $\underline{Y}(k^*) \neq \underline{0}$, this implies that

$$_i\underline{Y}_j(k^*)(\underline{e}_j - \underline{e}_i) \leq 0 \quad \begin{array}{l}\text{for all } i \neq j \\ \text{for all } j\end{array}$$

hence, from (5.48), (5.44), and (5.51), (5.55) and (5.57), one has

$$\underline{H}[\underline{Y}(k)] = \underline{0} \quad \text{for all } k \geq k^*$$
$$\underline{B}(k) = \underline{B}(k^*) \quad \text{for all } k \geq k^*$$
$$\underline{U}(k) = \underline{U}(k^*) \quad \text{for all } k \geq k^*$$
$$\underline{Y}(k) = \underline{Y}(k^*) \quad \text{for all } k \geq k^*$$

and

$$\Delta V[\underline{Y}(k)] = 0 \quad \text{for all } k \geq k^*$$

This completes the proof of Part 2.

Therefore, the algorithm (5.51), together with p(k) given by equation (5.52) under the condition (5.53) and with $\underline{B}(0)$ given by equation (5.54), is a convergent algorithm for the solution $\underline{U}$ of the set of linear inequalities (5.19). The nonlinear separability of the multiclass patterns can also be detected by observing at a certain step k^*

$$\underline{Y}_j(k^*)\,(\underline{e}_j - \underline{e}_i) \leq 0 \quad \begin{array}{l}\text{for all } i \neq j \\ \text{for all } j = 1,2,\ldots,R.\end{array}$$

VI. SUMMARY AND CONCLUSION

In this dissertation, a new iterative algorithm has been developed to solve for a solution $\underline{w}$, if one exists, to a set of linear inequalities, $\underline{A}\ \underline{w} > \underline{0}$ which arises in pattern dichotomization and switching problems. It is an improvement of the Ho-Kashyap algorithm based upon the attempt to minimize a different criterion function $J(\underline{y}) = 4 \sum_{i=1}^{N} (\cos \frac{1}{2} y_i)^2$ where $\underline{y} = \underline{A}\ \underline{w} - \underline{b}$ and $\underline{b}$ is a vector with all positive components. This criterion function has a larger gradient than the one used by Ho and Kashyap. The algorithm is expressed in equation (2.29) with the incremental coefficient p(k) given by either equation (2.47) or equation (2.26). The algorithm also simultaneously tests for the nonexistence of a solution of the linear inequalities whenever $\underline{y} \leq \underline{0}$.

This algorithm has a higher rate of convergence than previous methods for a certain range of the choice of $\underline{b}(0)$. A comparison has been made between this improved algorithm with p(k) given by equation (2.47) and the Ho-Kashyap algorithm with $p=1$, the convergence rate may be greatly increased for $.001 \leq b_i(0) \leq 0.5$ $(i=1,2,..,N)$, as verified by the computer results of switching theory and pattern classification problems in Chapters III and IV. For problems where a large number of iterations, for example, greater than twenty, were

required for the Ho-Kashyap algorithm, the proposed algorithm reduced this number of iterations by a factor of 20 to 450. For problems where a small number of iterations were required by the Ho-Kashyap algorithm, for example, less than twenty, the proposed algorithm reduced the number of iterations by as much as 30 percent.

The generalization of the proposed algorithm for a solution matrix $\underline{U}$ of a set of linear inequalities $\underline{A}_j\underline{U}(\underline{e}_j-\underline{e}_i) > \underline{0}$, (for all $i \neq j$ and $j=1,2,...,R$), which is applicable to multclass pattern classification has been presented and a convergence proof has been given. This generalized algorithm is expressed in equation (5.51) with p(k) given by equation (5.52). The convergence proof utilizes the concept of mapping the pattern classes into vertices of an equilateral simplex whose vertex vectors are $\underline{e}_i$, $(i=1,2,...,R)$.

The following six problems are suggested for further investigations: (1) to study in detail the relationship between the rate of convergence of the algorithm and the choice of p(k) and $\underline{b}(0)$; (2) to incorporate the proposed algorithm into the group-pattern adaptive procedure for pattern classification; (3) to apply the proposed algorithm for the development of an algorithm for piecewise linear separation in cases where the sample patterns are not linearly separable; and (4) to develop explicit algorithms to solve for nonlinear discriminant functions for some nonlinearity separable pattern recognition problems; (5) to extend the algorithm so that it can assure all $w_i > 0$,

($i=1,2,\ldots,n$), for threshold logic circuits realizable by transistors; and (6) to develop a procedure to select an adequate set of pattern attributes providing for reliability and flexability in a teaching machine.

APPENDIX A

PROGRAM LISTING FOR THE SPECIAL ALGORITHM OF EQUATION (3.5)

A MAD Program listing is shown on the following pages for the application of the accelerated algorithm to switching functions. The program is devised so that the iterations for various initial values of the b vector, b(0), can be performed successively. This is done by inputing NVAL equal to the number of initial b vectors and VALU (1)...VALU(NVAL) equal to the initial values of the b vector. The matrix A is read in by FORMAB. FORMAB inputs

ID = identification number

N = number of columns of A

M = number of rows of A

NA = number of elements of class 1

MB = number of elements of class 2

and the elements of class 1 and class 2 with the minterm expressed in decimal form, elements of class 1 are entered first. The value of p(k) in equation (2.47) is used for the program listing.

```
$COMPILE MAD,EXECUTE,DUMP
             R   ///        ALGORITHM FOR LINEAR INEQUALITIES        ///
              D'N       A(100*30),Z(100),      B(100),W(30),Y(100),SPMS(100)
              DIMENSION VALU(100)
              DIMENSION NA(500)
              PROGRAM COMMON ID,N,M,A
              INTEGER I,J,K,M,N,ALLY,L,NVAL,R
              READ FORMAT FVAL, NVAL,VALU(1)...VALU(NVAL)
BGIN          EXECUTE FORMAB.
              SETDIM. (A,M,N)
              THROUGH STOP, FOR R=1,1,R .G. NVAL
             R    / FORM B MATRIX
              EXECUTE SPRAY. (VALU(R),B(1)...B(M))
              P'T FOUT, ID,(I=1,1,I .G. M, (J=1,1,J .G. N, A(I,J)),B(I))
             R    / 1.   W=1/2**(N-1)*(A'*B)
             R    /     W(J)=1/2**(N-1)*(A(1,J)*B(1)+A(2,J)*B(2)
             R    /                  +...+A(M,J)*B(M))          J=1,....,N
              THROUGH STEP1, FOR J=1,1,J .G. N
              W=0.
              THROUGH STEP1A, FOR I=1,1,I .G. M
STEP1A        W = W + A(I,J)*B(I)
STEP1         W(J) = W / 2. .P. (N-1)
             R    / K IS ITERATION NUMBER
              THROUGH DONE, FOR K=0,1,K .G.1000
             R    / 2.   Z=A*W
             R    /     Z(I)=A(I,1)*W(1)+A(I,2)*W(2)+...+A(I,N)*W(N)
             R    /                                               I=1,....,M
              ALLY=0
              THROUGH STEP2, FOR I=1,1,I .G. M
              Z(I)=0
              THROUGH STEP2A, FOR J=1,1,J .G. N
STEP2A        Z(I)=Z(I)+A(I,J)*W(J)
STEP2         WHENEVER Z(I) .G. 0., ALLY=ALLY+1
              P'T FALLY, Z(1)...Z(M)
              P'T FY, ALLY, -ALLY
             R    / 3. IF ALL Z(I) .G. 0 FOR I=1,....,M
             R    /     THEN PRINT OUT ''THE SOLUTION OF A*W .G. 0 IS''
             R    /     THEN PRINT OUT ALL W(J) FOR J=1,....,N
             R    /     ALSO PRINT OUT K AND STOP
              WHENEVER ALLY .E. M
                   PRINT FORMAT FSOLN, W(1)...W(N),K
                   TRANSFER TO STOP
              END OF CONDITIONAL
             R    /     THEN MODIFY Z SUCH THAT Y=A*W-B
             R    /     Y(I)=Z(I)-B(I)
              ALLY = 0
              THROUGH STEP3, FOR I=1,1,I .G. M
```

```
               Y(I)=Z(I)-B(I)
STEP5          WHENEVER Y(I) .G. 0, ALLY=ALLY+1
              R    / 4. IF ALL Y(I) .LE. 0 FOR I=1,...,M
              R    /    THEN PRINT OUT ''PROBLEM IS NOT LINEARLY SEPARABLE''
              R    /    ALSO PRINT OUT K AND STOP
               WHENEVER ALLY .E. 0
                    PRINT FORMAT FNOSOL, K
                    TRANSFER TO STOP
               END OF CONDITIONAL
              R    / 7.  CALCULATE (S+.ABS.S) = SPMS
              R    /    SPMS(I)=1/2*(EXP(  Y(I))-EXP(-  Y(I)) +
              R    /          .ABS.(EXP(  Y(I))-EXP(-  Y(I)))   I=1,....,M
               THROUGH STEP7, FOR I=1,1,I .G. M
               SPMS = EXP.(  Y(I))-EXP.(-  Y(I))
STEP7          SPMS(I) = (SPMS + .ABS. SPMS)/ 2.
              R / SET P= (  Y+.ABS.  Y)'*SPMS/SPMS'*(I- A*A')/2**(N-1))*SPMS
              R
              R    / CALCULATE NUMERATOR
               NUMP=0.
               THROUGH NUM, FOR I=1,1,I .G. M
NUM            NUMP=NUMP +(  Y(I)+.ABS.  Y(I))*SPMS(I)
              R    / CALCULATE DENOMINATOR
               BOTM=0.
               THROUGH STEP6, FOR I=1,1,I .G. M
               THROUGH STEP6B, FOR L=1,1,L .G. M
               NA(L) = 0.
               THROUGH STEP6A, FOR J=1,1,J .G. N
STEP6A         NA(L) = NA(L) - A(L,J)*A(I,J)
               NA(L) = NA(L)/2. .P. (N-1)
STEP6B         WHENEVER I .E. L, NA(L) = NA(L) + 1.
               DENOM=0.
               THROUGH DEN, FOR J=1,1,J .G. M
DEN            DENOM = DENOM + SPMS(J)*NA(J)
STEP6          BOTM = BOTM + DENOM*SPMS(I)
               P = NUMP/(BOTM*2.)
               P'S P
              R    / 8.  THEN MODIFY B SUCH THAT B = B + P*SPMS
              R    /    B(I) = B(I) + P*SPMS(I)                    I=1,....,M
               THROUGH STEP8, FOR I=1,1,I .G. M
STEP8          B(I) = B(I) + P*SPMS(I)
              R    / 9.  MODIFY W SUCH THAT W=W + 1/2.P.(N-1)*P*A'*SPMS
              R    /    W(J) = W(J) + 1/2.P.(N-1)*P*(A(1,J)*SPMS(1)+A(2,J)*SPMS(2)
              R    /          +...+A(M,J)*SPMS(M))                 J=1,....,N
               THROUGH STEP9, FOR J=1,1,J .G. N
X              W=0.
               THROUGH STEP9A, FOR I=1,1,I .G. M
STEP9A         W=W+ A(J,I)*SPMS(I)
```

```
STEP9        W(J)=W(J)+P/2.P.(N-1)*W
            R     / RETURN TO STEP 2
DONE
            R     / IF K.G.1000 PRINT OUT ''K IS GREATER THAN 1000
             PRINT COMMENT $-  K IS GREATER THAN 1000$
STOP         CONTINUE
             TRANSFER TO BGIN
             FORMAT VARIABLE N
             VECTOR VALUES FVAL=$I3,7(F10.8)*$
             V'S FOUT=$H+1PROBLEM +C2//(S2,'N'(S2,F12.6),S5,F8.4)*$
             VECTOR VALUES FSOLN=$H+-  THE SOLUTION OF A*W .G. 0 IS +/
            1'N'(S2,F9.4)/4H K= I4*$
             VECTOR VALUES FNOSOL=$H+-  PROBLEM IS NOT LINEARLY SEPARABLE
            1+/4H K= I4*$
             V'S FALLY=$1H0/(10E13.6)*$
             V'S FY=$H+-THE NUMBER OF POSITIVE Y(I)=A(I,J)*W(J) IS +I5//
            1H+THE NUMBER OF NONPOSITIVE Y(I)=A(I,J)*W(J) IS +I5*$
             END OF PROGRAM
$ASSEMBLE
        ENTRY    FORMAB
FORMAB  SXA      XR4,4
* READ  IN DATA
        READ     FMT,ID,N,M,NA,NB
        CLA      M
        ALS      18
        STD      INST
        AXT      1,1
LOOP    IOH      PART,1                  READ DATA
        TXI      *+1,1,1
INST    TXL      LOOP,1,**
        ENDIO
* CONVERT DATA FOR PA PARTITION FOR A MATRIX
        CLA      N
        SUB      =1
        STA      N4
        STA      N1
        AXT      1,1
        AXT      1,4
LOOP3   LXA      N4,2
        CLA      =1.
        STO      A,4
        TXI      *+1,4,1
BELOW   CLA      PART,1
N4      LRS      **
BACK2   ZAC
        LLS      1
        SXA      XR2,2
```

```
         PAX      0,2
         CAL      ATABLE,2
         SLW      A,4
         TXI      *+1,4,1
XR2      AXT      **,2
         TIX      BACK2,2,1
         TXI      *+1,1,1
         CLA      NA
         SUB      =1
         STO      NA
         TNZ      LOOP3
* CONVERT DATA    FOR PB PARTITION FOR A MATRIX
LOOP2    LXA      N4,2
         CLA      =-1.
         STO      A,4
         TXI      *+1,4,1
BELOW2   CLA      PART,1
N1       LRS      **
BACK     ZAC
         LLS      1
         SXA      XR22,2
         PAX      0,2
         CAL      BTABLE,2
         SLW      A,4
         TXI      *+1,4,1
XR22     AXT      **,2
         TIX      BACK,2,1
         TXI      *+1,1,1
         CLA      NB
         SUB      =1
         STO      NB
         TNZ      LOOP2
XR4      AXT      **,4
         TRA      1,4
         PCLIST   ID,N,M
A        PGMCOM   10000
NA       BSS      1
NB       BSS      1
PART     BES      2500
         DEC      1.                     1 CONVERTS TO 1
ATABLE   DEC      -1.                    0 CONVERTS TO -1
         DEC      -1.                    1 CONVERTS TO -1
BTABLE   DEC      1.                     0 CONVERTS TO 1
FMT      BCI      *,C2,S1,I2,3I3,/19(I4)*
         END
```

APPENDIX B

PROGRAM LISTING FOR THE GENERALIZED INVERSE OF A MATRIX

A program listing in MAD language is shown on the following pages for the calculation of the generalized inverse of a matrix. The program is written for M = number of rows greater than N = number of columns. The matrix $\underline{A}$ is read in by rows, a_{11} first.

```
$ COMPILE MAD, EXECUTE, DUMP
           PROGRAM COMMON A,AI,M,N
           D'N A(100*30),AI(30*100),AT(30*100),B(30*30),C(30*30),
          1D(30*30),T(30*30),U(30*30),FMT(12)
           INTEGER I,JC,J,KA,KB,K,KT,L,LT,LX,MX,NX
           INTEGER M,N
           R'T F100,M,N
           R'T F200, (J=0,1,J.G.11,FMT(J))
           R'T FMT, (I=1,1,I.G.M,(J=1,1,J.G.N,A(I,J)))
           KA=N
           KB=M
           P'T FMT2,(I=1,1,I.G. M, (J=1,1,J.G.N, A(I,J)))
           FORMAT VARIABLE M,N
           V'S FMT2=S'N'F9.3*S
           T'H L20,FOR I=1,1,I .G. KB .
           T'H L20, FOR J=1,1,J .G. KB
L20        B(I,J)=0.
           T'H L22, FOR J=1,1,J .G. KB
           THROUGH L22, FOR I=1,1,I .G. KA
L22            AT(I,J)=A(J,I)
           THROUGH L23, FOR I=1,1,I .G. KA
           T'H L23, FOR J=1,1,J .G. KA
           T'H L23, FOR K=1,1,K .G. KB
L23        B(J,I)=B(J,I)+AT(J,K)*A(K,I)
           T'H L21, FOR I=1,1,I .G. KA
           T'H L21, FOR J=1,1,J .G. KA
L21        D(I,J)=B(I,J)
           EXECUTE ADE.(B,B,T,KA,L)
           P'T F100,L
           W'R L .GE. KA, T'O T16
          RCONSTRUCT U MATRIX
           T'H L10, FOR I=1,1,I .G. KA
           T'H L10, FOR J=1,1,J .G. KA
           W'R B(I,I) .LE. 0
                     U(I,J)=-T(I,J)
           O'E
                     U(I,J)=0.
           E'L
L10        U(I,I)=B(I,I)
          RCOMPUTE  UTGU=C
           JC=2
           LT=1
           EXECUTE TMULT. (U,D,C,KA,JC,LT,KA,KA,KA)
          RSET ROWSCOL OF C=0
           T'H L12, FOR I=1,1,I .G. KA
           W'R  B(I,I) .E. 0.
           T'H L12, FOR J=1,1,J .G. KA
           C(I,J)=0.
L12        C(J,I)=0.
```

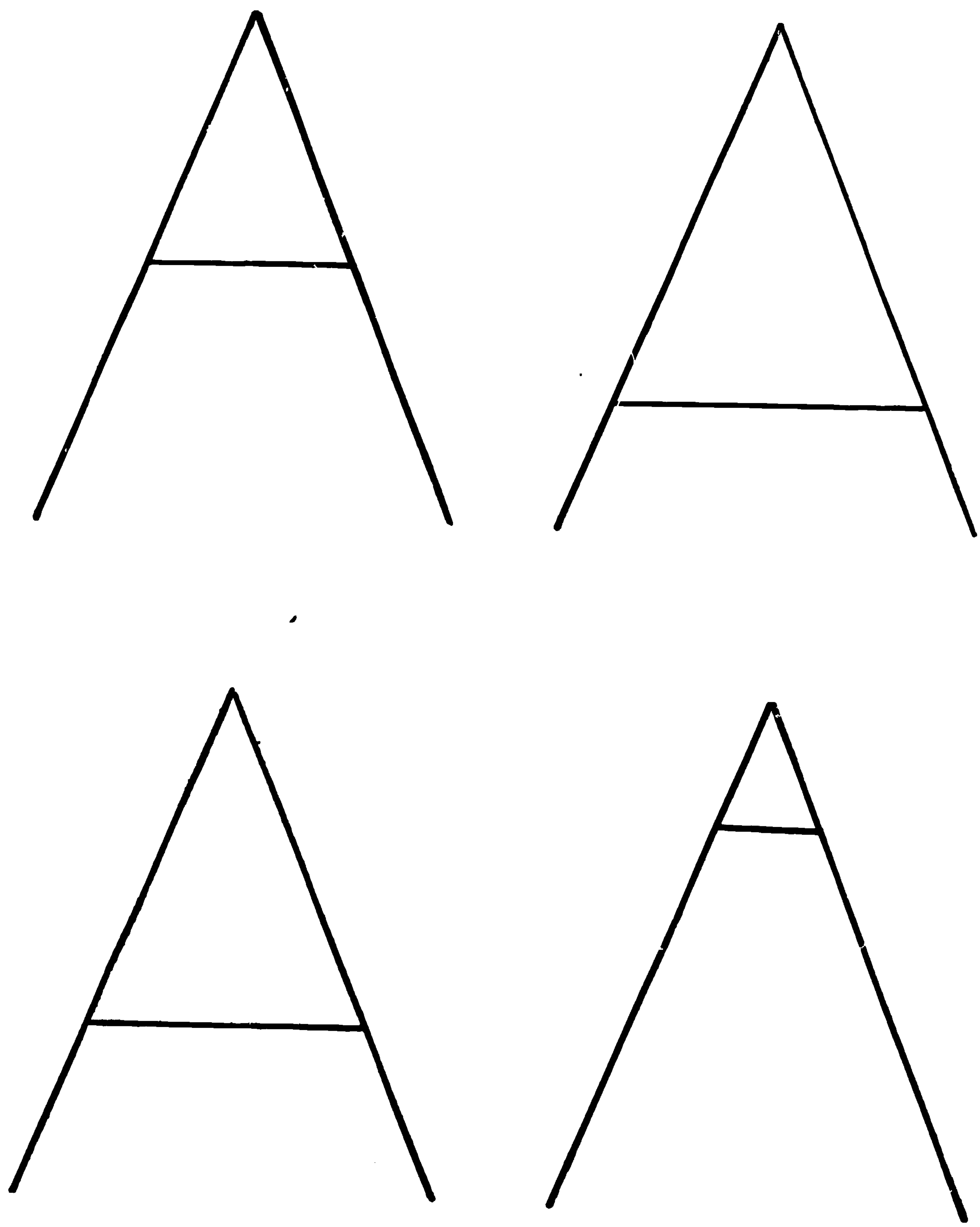
```
            END OF CONDITIONAL
L12         CONTINUE
           REXTRACT D MATRIX FROM C
            NX=0
            T'H L15, FOR I=1,1, I.G.KA
            T'H L15, FOR J=1,1, J .G. KA
            W'R C(J,I) .NE. 0.
            NX=NX+1
            MX=0
            T'H L15, FOR K=J,1, K .G. KA
            W'R C(K,I) .NE. 0
            MX=MX+1
            D(MX,NX)=C(K,I)
            C(K,I)=0.
            E'L
L15         E'L
            EXECUTE ADE. (D,C,T,MX,L)
            LX=0
            T'H L37, FOR I=1,1,I .G. MX
T30         LX=LX+1
            W'R B(LX,LX) .LE. 0
            T'H L32, FOR KT=1,1,KT .G. KA
L32         T(LX,KT)=0.
            W'R LX .L. KA, T'O T30
            T'O L37
            E'L
            K=0
            T'H L37, FOR J=1,1,J .G. MX
T34         K=K+1
            W'R B(K,K) .LE. 0
            T(LX,K)=0.
            W'R K .L. KA, T'O T34
            T'O L37
            E'L
            T(LX,K)=C(I,J)
L37         CONTINUE
           RCONSTRUCT UDUT
            JC=1
            LT=1
            EXECUTE TMULT. (U,T,B,KA,JC,LT,KA,KA,KA)
           RNORM ERROR CONTOL NOT INCLUDED
           RCOMPUTE INVES
T16         T'H L17, FOR I=1,1,I .G. KA
            THROUGH L17, FOR J=1,1,J .G. KB
            AI(I,J)=0.
            T'H L17, FOR K=1,1,K .G. KA
L17         AI(I,J)=AI(I,J) + B(I,K)*AT(K,J)
            P'T FMT3, (I=1,1,I.G.KA,(J=1,1,J.G.KB, AI(I,J)))
            V'S FMT3=$'M'F12.6*$
```

```
              X=B(K,K)
              IK=K
L10           END OF CONDITIONAL
              NS(IK)=IK
              X=1./SQRT.(X)
              T'H L11, FOR J=1,1,J .G. KA
              W'R IK .NE. J
              B(IK,J)=B(IK,J)*X
              B(J,IK)=B(J,IK)*X
              T(IK,J)=T(IK,J)*X
              T(J,IK)=T(J,IK)*X
L11           END OF CONDITIONAL
              T(IK,IK)=T(IK,IK)*X
              B(IK,IK)=1.0
              T'H L13, FOR I=1,1,I .G. KA
              T'H L13, FOR J=1,1,J .G. KA
L13           W'R I .NE. IK .AND. J .NE. IK, B(I,J)=B(I,J)-B(IK,J)*B(I,IK)
              T'H L21, FOR I=1,1,I .G. KA
              T'H L21, FOR J=1,1, J .G. KA
L21           W'R I .NE.IK, T(I,J)=T(I,J)-T(IK,J)*B(I,IK)
              T'H L14, FOR J=1,1,J .G. KA
              B(J,IK)=0.
L14           B(IK,J)=0.
              B(IK,IK)=1.
              MT=0
              L=0
              T'H L17, FOR J=1,1,J .G. KA
              W'R B(J,J) .LE. .0001
              B(J,J)=0.
              MT=MT+1
              OR WHENEVER B(J,J) .E. 1.
              MT=MT+1
              L=L+1
L17           END OF CONDITIONAL
              WHENEVER MT .E. KA, TRANSFER TO T3
L69           CONTINUE
             RCHECK RANK
T3            W'R L .E. KA
              LT=2
              JC=2
              EXECUTE TMULT. (T,T,E,KA,JC,LT,KA,KA,KA)
              END OF CONDITIONAL
              FUNCTION RETURN
              END OF FUNCTION
```

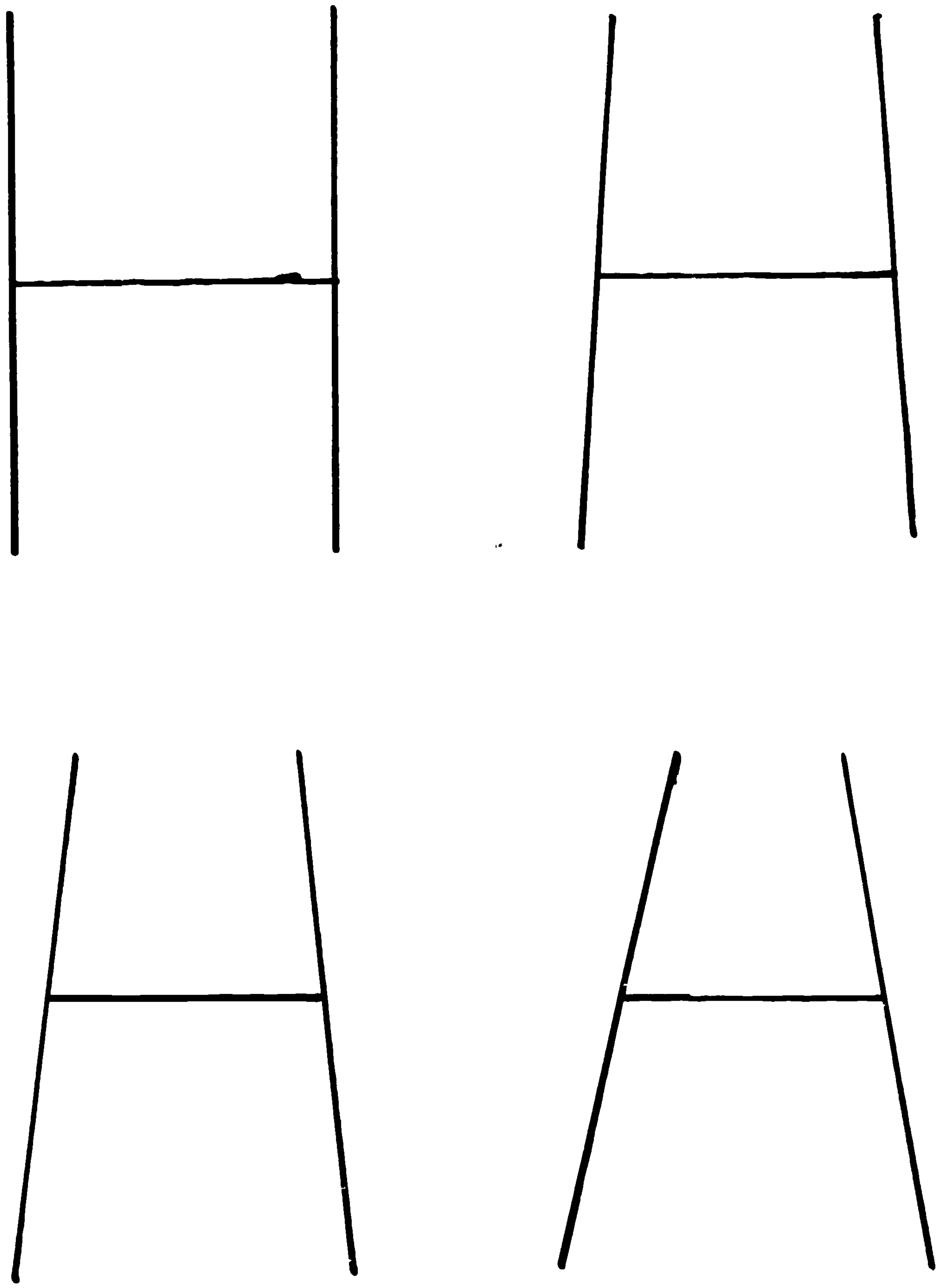
```
              V'S F100=$2I3*$
              V'S F200=$12A6*$
              END OF PROGRAM
$COMPILE MAD
              EXTERNAL FUNCTION (A,B,C,KA,JC,LT,KB,KC,KD)
              ENTRY TO TMULT.
             RA=MAT TO BE TRANS.  C=PROD.  JC=1-BAT  J=2-ATB
             RLT=1-ABAT OR ATBA, LT=I-BAT OR ATB
              DIMENSION (T,U) (30*30)
              INTEGER I,J,K, JC,KA,KB,KC,KD,LT
              T'H L1, FOR J=1,1,J .G. KB
              T'H L1, FOR I=1,1,I .G. KA
              C(I,J)=0.
              U(J,I)=B(J,I)
L1            T(I,J)=A(J,I)
T1            T'H L4, FOR I=1,1,I .G. KA
              T'H L4, FOR J=1,1,J .G. KC
              T'H L4, FOR K=1,1,K .G. KD
              T'O L(JC)
L(1)          C(I,J)=C(I,J) + U(I,K)*T(K,J)
              T'O L4
L(2)          C(J,I)=C(J,I) + T(J,K)*U(K,I)
L4            CONTINUE
              T'O T57(LT)
T57(1)        THROUGH L6, FOR I=1,1,I .G. KA
              T'H L6, FOR J=1,1,J .G. KA
              T(I,J)=C(I,J)
              U(I,J)=A(I,J)
              C(I,J)=0.
L6            LT=2
              T'O T1
T57(2)        CONTINUE
              FUNCTION RETURN
              END OF FUNCTION
$COMPILE MAD
              EXTERNAL FUNCTION (B,E,T,KA,L)
              ENTRY TO ADE.
              INTEGER IK,I,JC,JI,J,K,LSI,LT,MT,NS,KA,L
              DIMENSION NS(30)
              IK=0
              T'H L7, FOR I=1,1,I .G. KA
              NS(I)=0
              T'H L7, FOR J=1,1,J .G. KA
              T(I,J)=0.
L7            T(I,I)=1.
              T'H L69, FOR LSI=1,1,LSI .G. KA
              X=0.
              T'H L10, FOR K=1,1,K .G. KA
              W'R B(K,K) .GE. X .AND. NS(K) .L. K
```

APPENDIX C

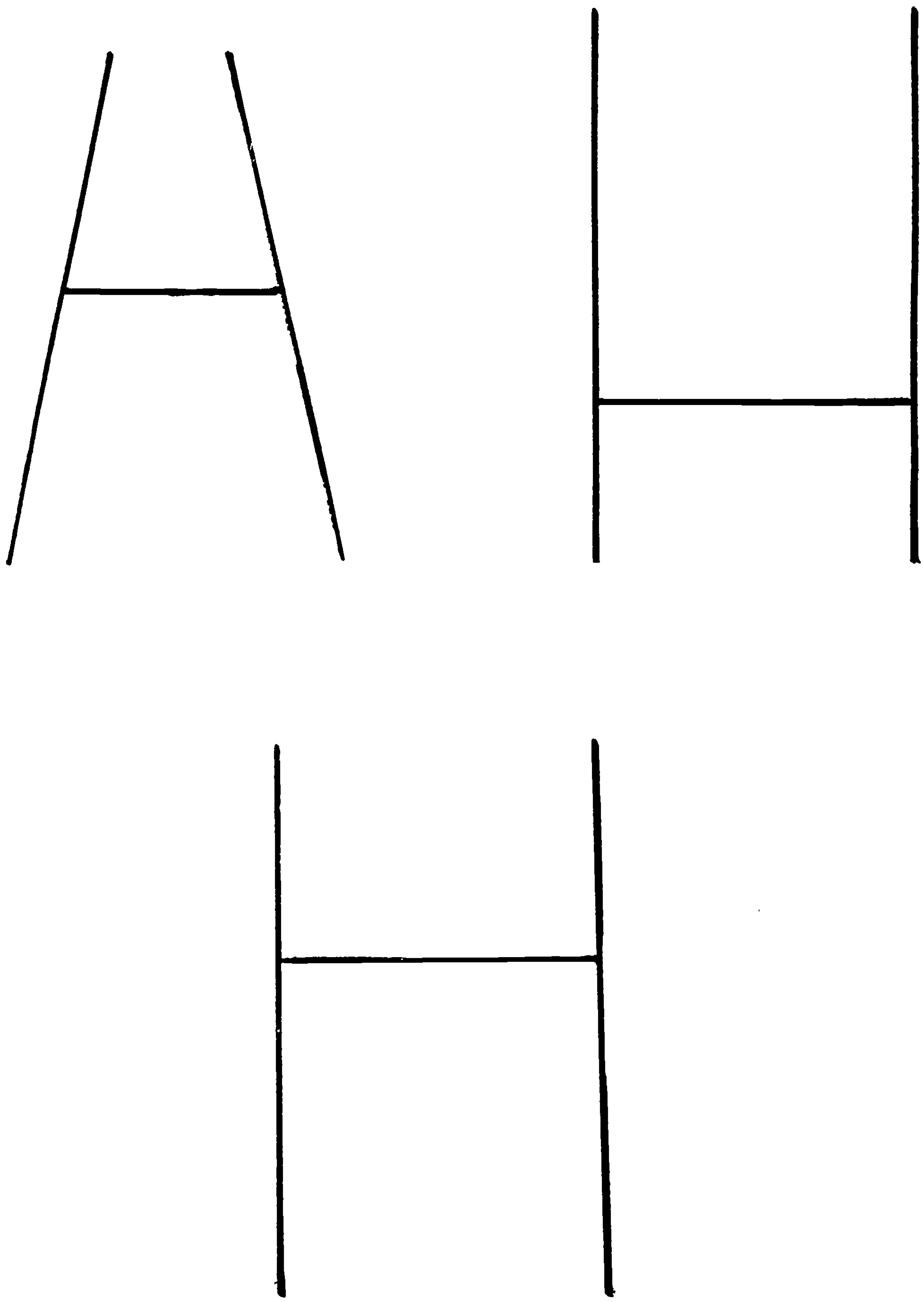
SAMPLE PATTERNS OF ALPHANUMERIC CHARACTERS WRITTEN BY CHILDREN



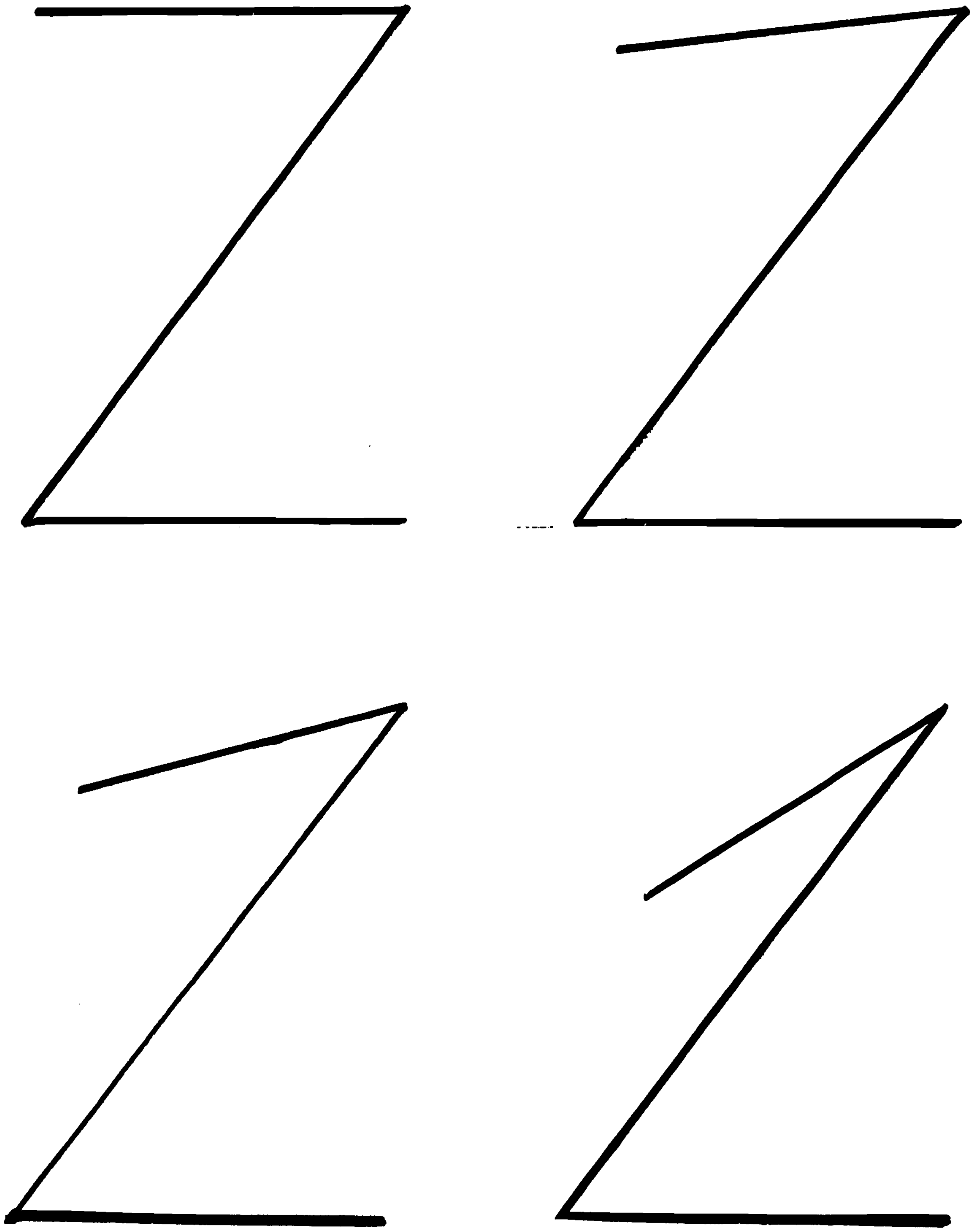
The letter A



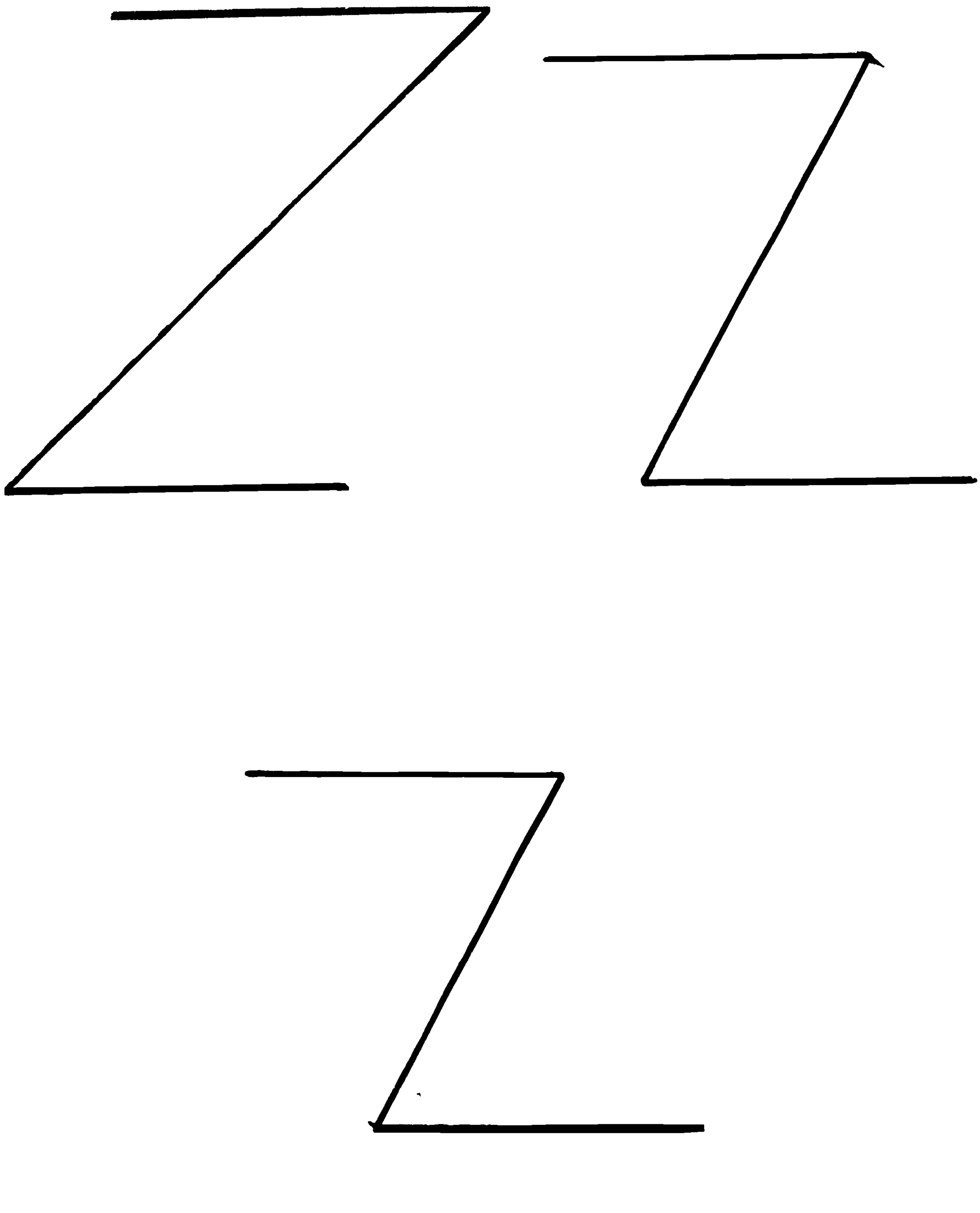
The letter H



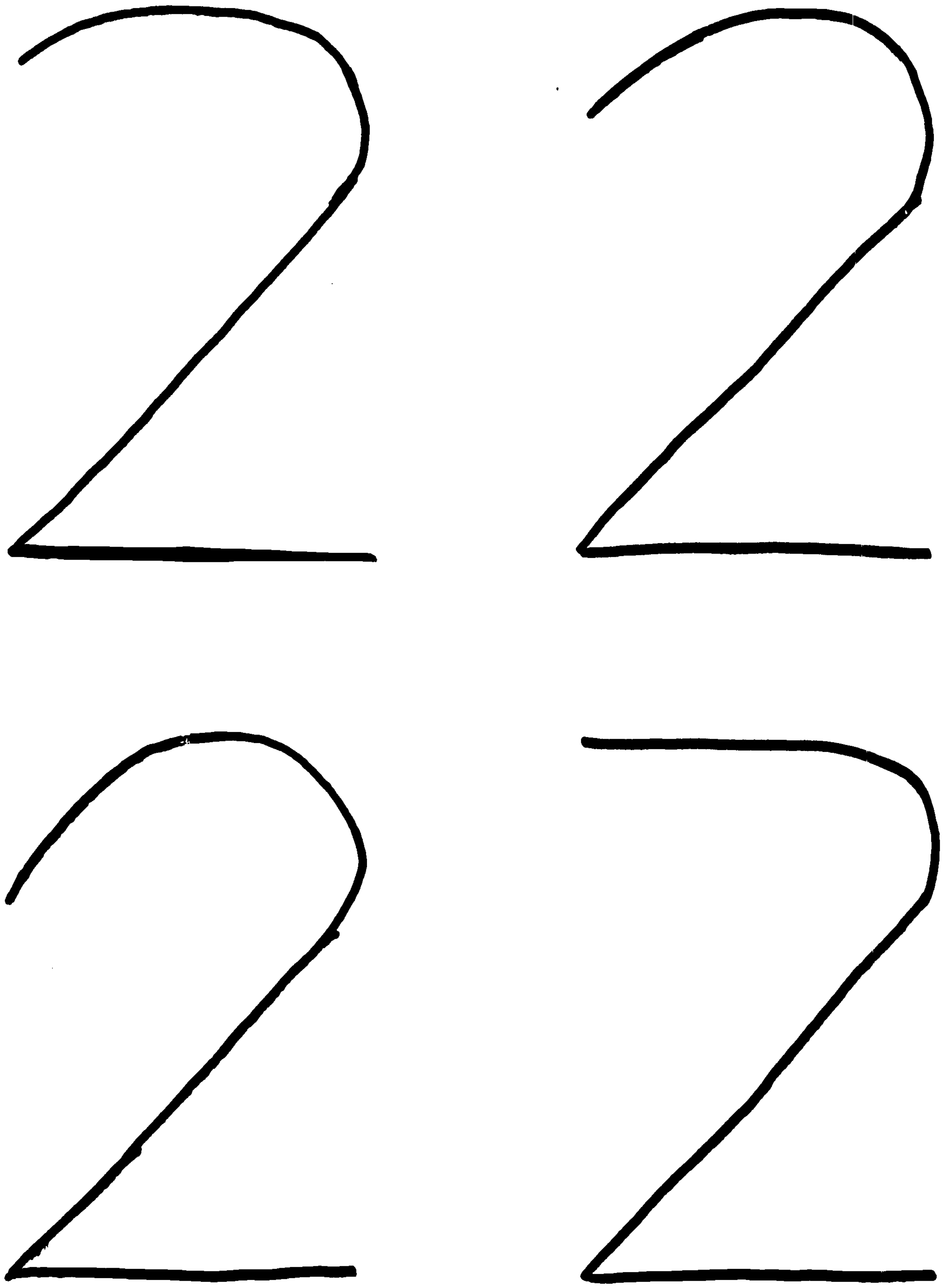
The letter H



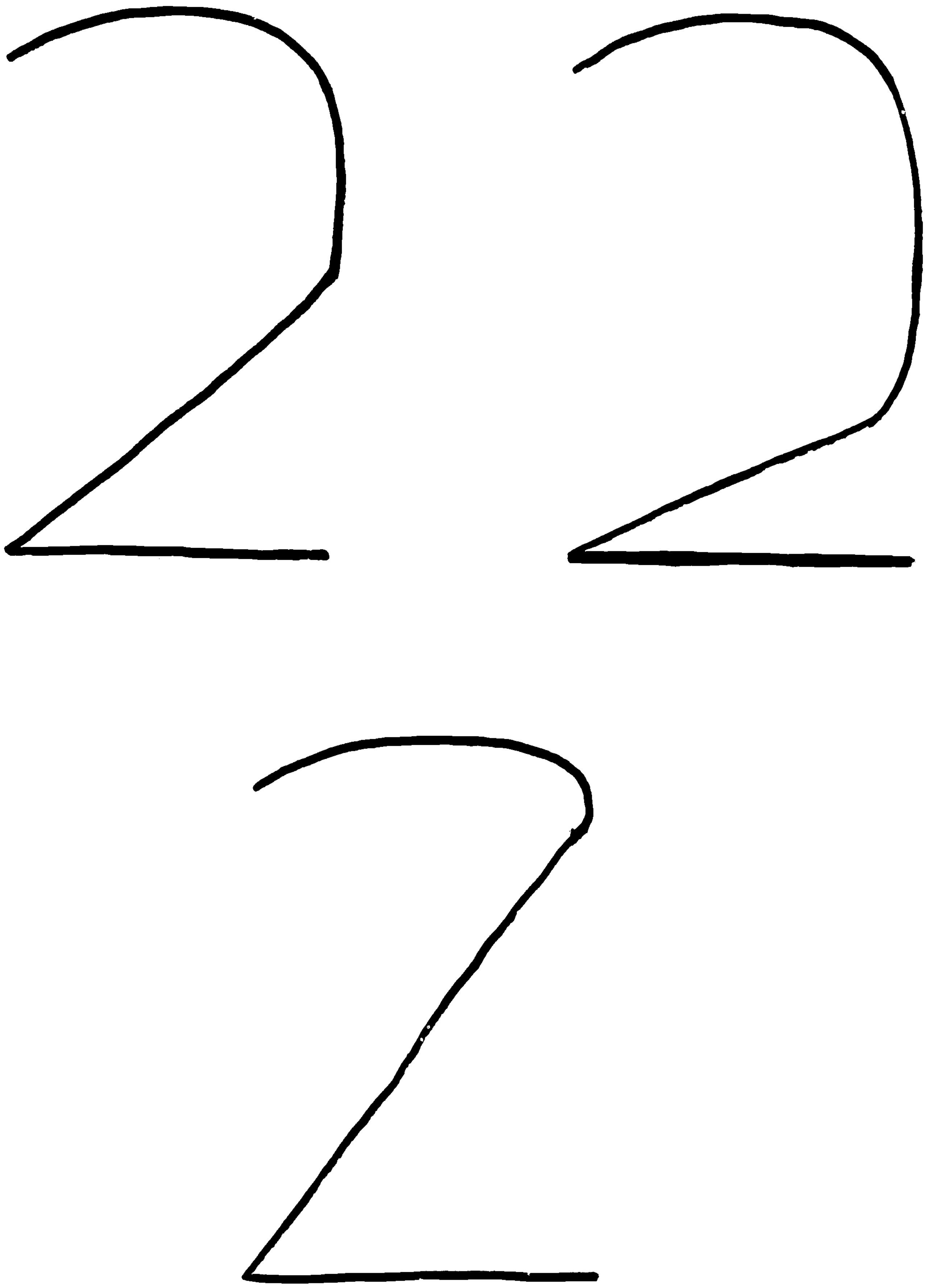
The letter Z



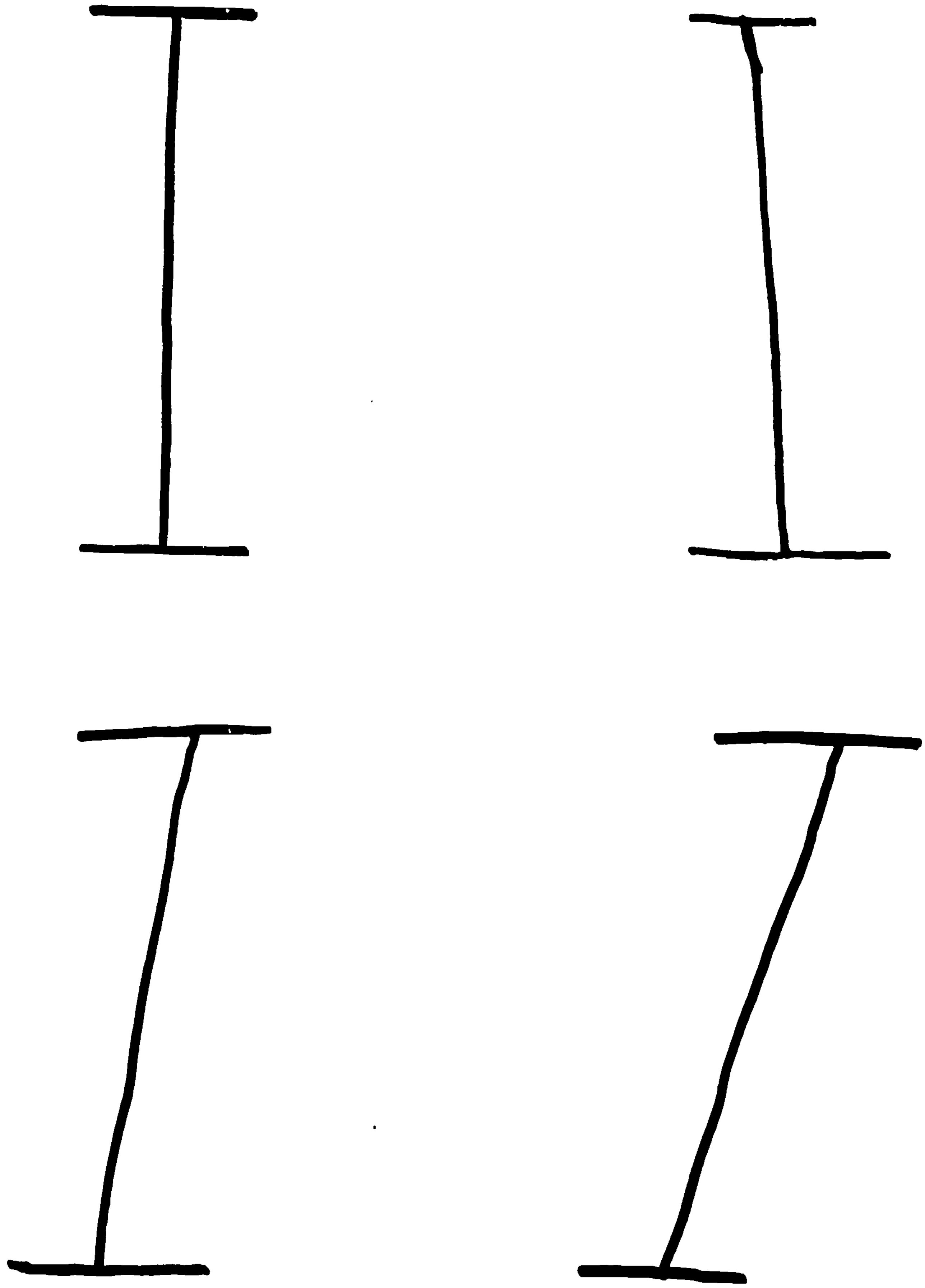
The letter Z



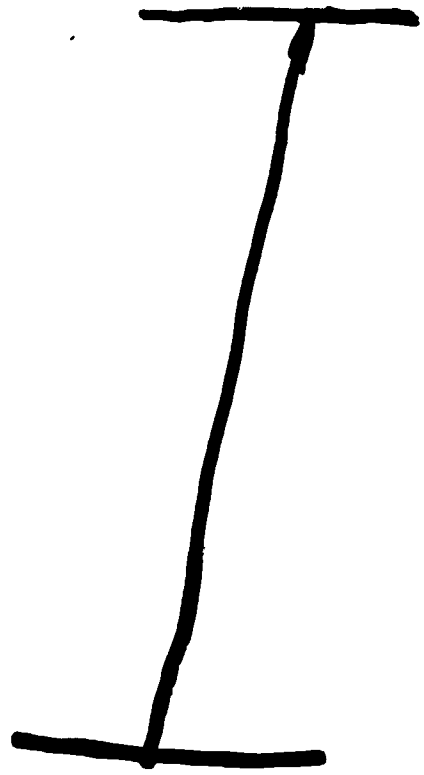
The number 2



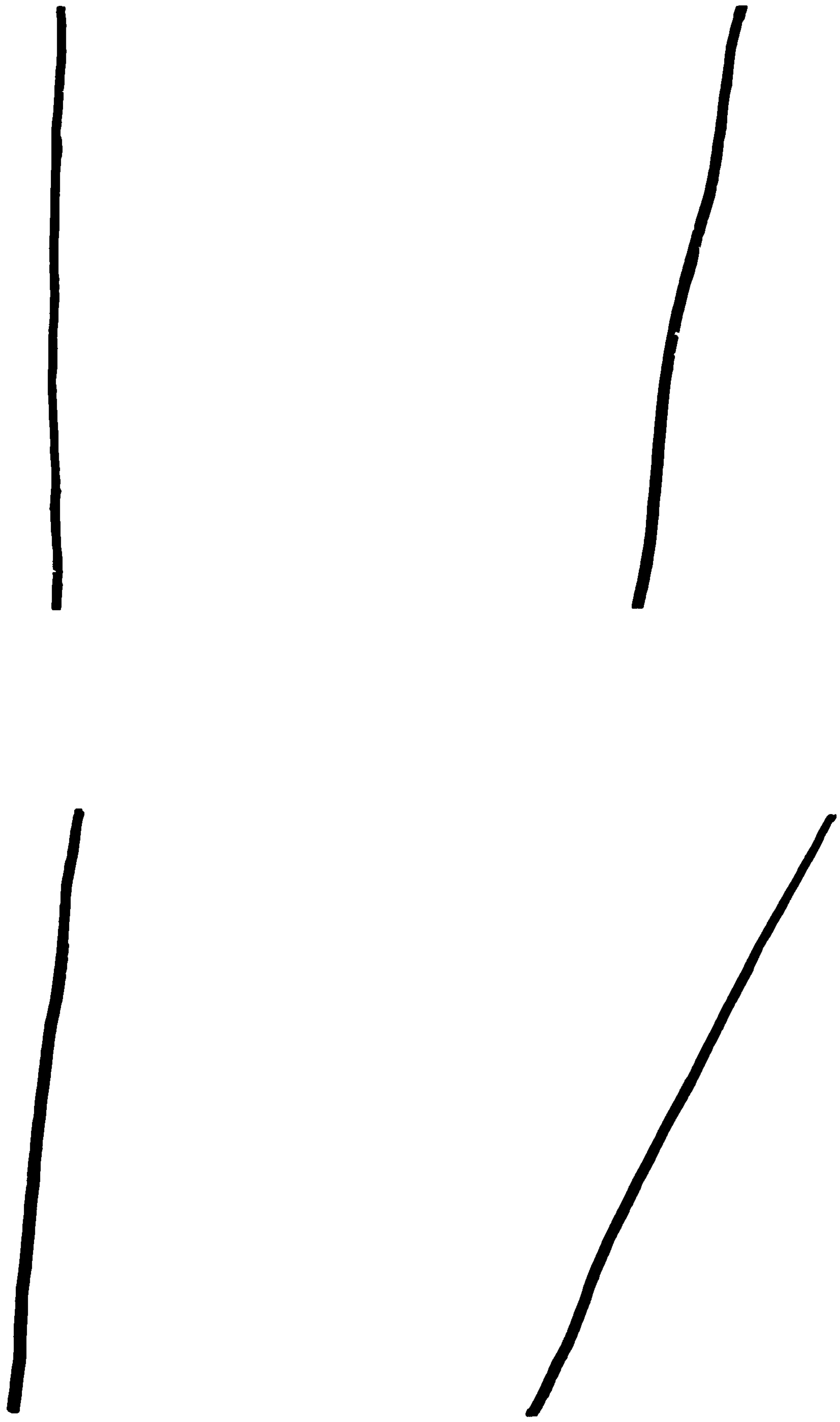
The number 2



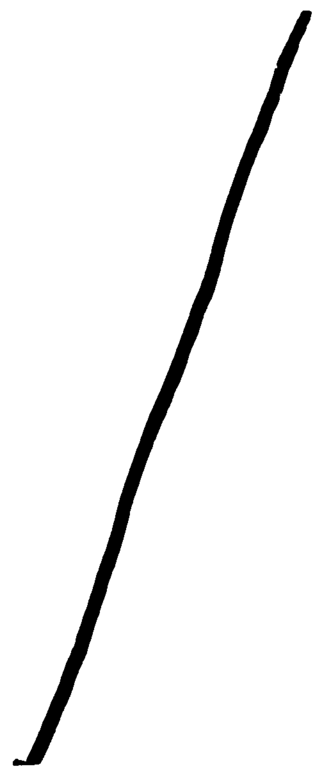
The letter I



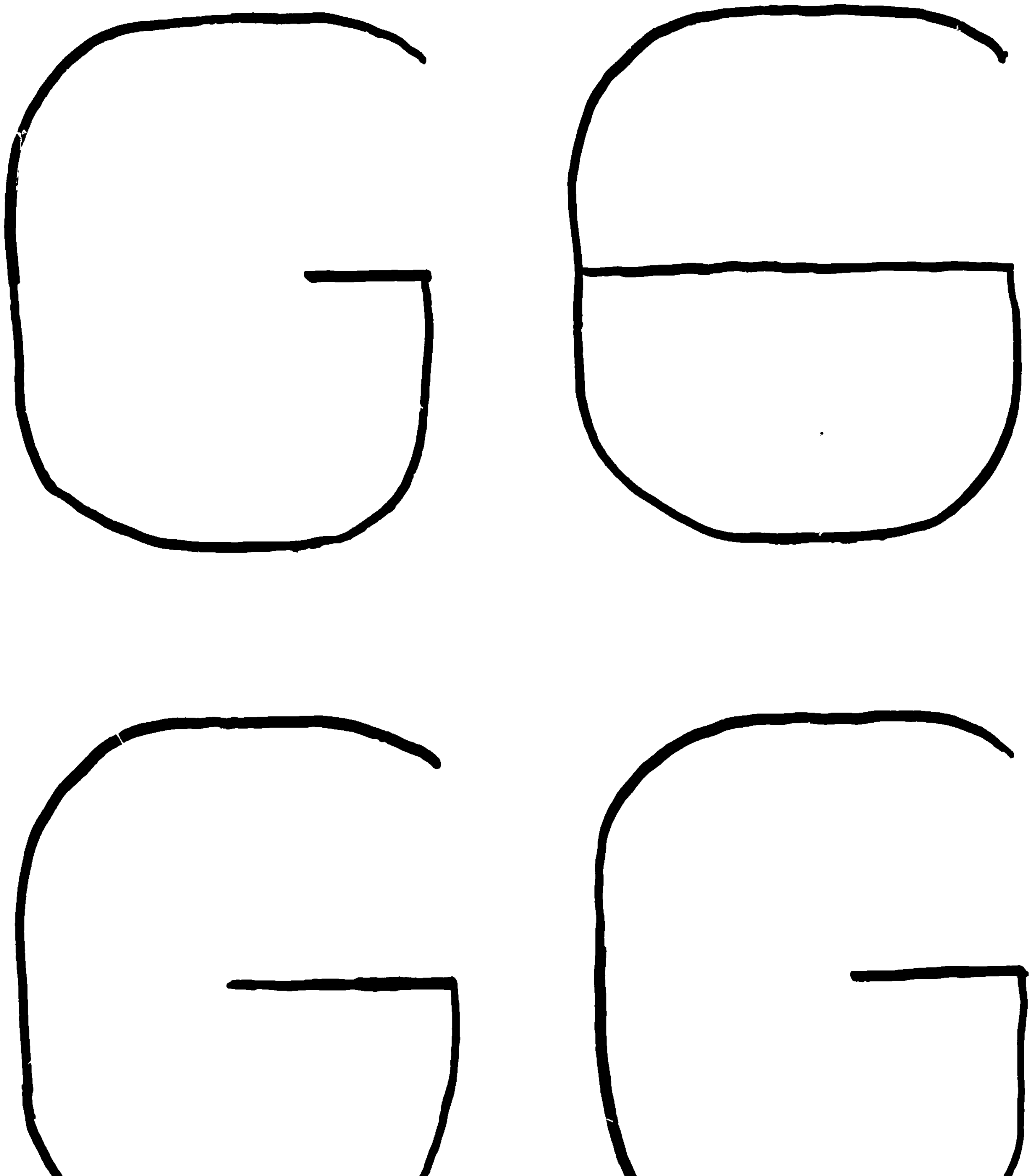
The letter I



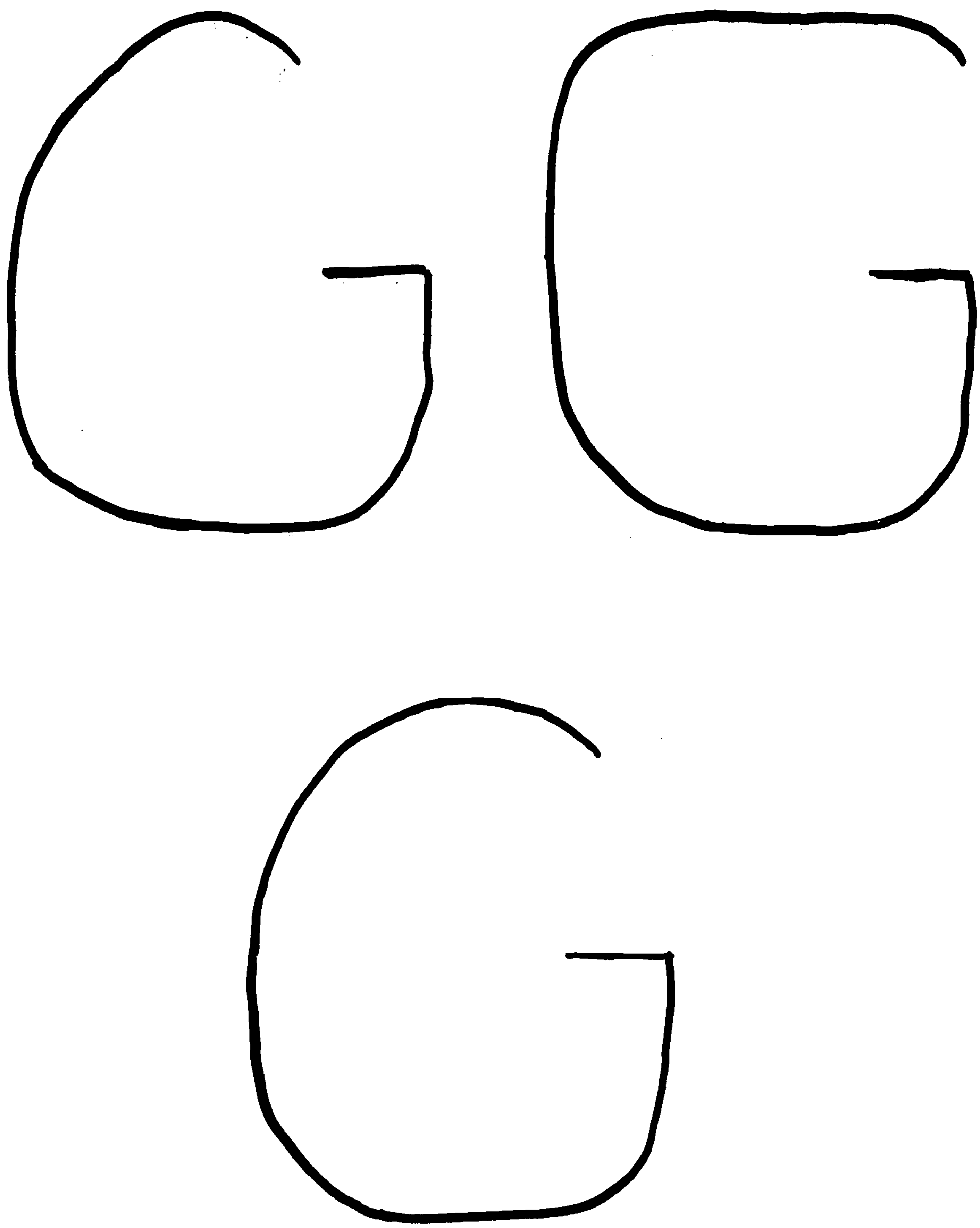
The number 1



The number 1



The letter G



The letter G

The number 6

BIBLIOGRAPHY

1. Nilsson, N. J. Learning Machines. New York: McGraw Hill, Inc., 1965.

2. Sebestyen, G. S. Decision-making Processes in Pattern Recognition. New York: MacMillan Co., 1962.

3. Arkadev, A. G. and E. M. Braverman. Computers and Pattern Recognition. Washington, D. C.: Thompson Book Co., 1967.

4. Uhr, L. Pattern Recognition. New York: John Wiley and Sons, Inc., 1966.

5. Chow, C. K., "An Optimum Character Recognition System Using Decision Functions," IRE Transactions on Electronic Computers, Vol. EC-6, pp. 247-254, 1957.

6. Winder, R. O., "Threshold Logic," Ph.D. Dissertation, Princeton University, 1962.

7. Lewis, P. M. I. and C. L. Coates. Threshold Logic. New York: John Wiley and Sons, Inc., 1967.

8. Caldwell, S. H. Switching Circuits and Logical Design. New York: John Wiley and Sons, Inc., 1958.

9. Phister, M., Jr. Logical Design of Digital Computers. New York: John Wiley and Sons, Inc., 1958.

10. Hoff, M. E. and B. Widrow, "Adaptive Switching Circuits," Stanford Electronics Laboratories, TR No. 1553-1, June, 1960.

11. Mays, C. H., "Adaptive Threshold Logic," Stanford Electronics Laboratories, TR No. 1557-1, April, 1963.

12. Ho, Y. C. and R. L. Kashyap, "An Algorithm for Linear Inequalities and its Applications," IEEE Transactions on Electronic Computers, Vol. EC-14, No. 5, pp. 683-688, October, 1965.

13. Kashyap, R. L., "Pattern Classification and Switching Theory," Ph.D. Dissertation, Harvard University, 1965.

14. Chaplin, W. G. and V. S. Levadi, "A Generalization of the Linear Threshold Decision Algorithm to Multiple Classes," Computer and Information Sciences - II, edited by J. Tou, New York: Academic Press, pp. 337-355, 1967.

15. Blaydon, C. C., "Recursive Algorithms for Pattern Recognition," Ph.D. Dissertation, Harvard University, 1967.

16. Fu, K. S. and W. G. Wee, "On Generalizations of Adaptive Algorithms and Application of the Fuzzy Sets Concept to Pattern Classification," TR-EE67-7, School of Electrical Engineering, Purdue University, June, 1967.

17. Wee, W. G., "Generalized Inverse Approach to Adaptive Multiclass Pattern Classification," Proceedings of the Second Annual Princeton Conference on Information Sciences and Systems, pp. 442-446, 1968.

18. Devyaterikov, I. P., A. I. Propoi, and Y. Z. Tsypkin, "Iterative Learning Algorithms for Pattern Recognition," Automation and Remote Control, Vol. 28, No. 1, pp. 108-117, January, 1967.

19. Smith, F. W., "Pattern Classifier Design by Linear Programming," IEEE Transactions on Computers, Vol.C-17, No.4, pp.367-372, April, 1968.

20. Penrose, R., "A Generalized Inverse for Matrices," 1955 Proceedings of the Cambridge Philosophical Society, pp. 406-413, 1955.

21. Walsh, J. L., "A Closed Set of Normal Orthogonal Functions," American Journal of Mathematics, Vol. 45, pp. 5-24, January, 1923.

22. Kalman, R. E. and T. S. Englar, "A Users' Manual for the Automatic Synthesis Program," NASA Contractor Report, NASA CR-475, Washington, D. C., June, 1966.

23. Li, C. C., et el, "On the Convergence Proof of the Generalization of the Ho-Kashyap Algorithm to Multiclass Pattern Classification", an unpublished technical report, 1968.

REFERENCES NOT CITED

Agmon, S., "The Relaxation Method for Linear Inequalities", Canadian Journal of Mathematics, Vol. 6, pp. 382-392, 1954.

Akers, S. B., Techniques of Adaptive Decision Making, G. E. Electronics Laboratory Technical Information Series R65ELS-12, 1965.

Bodewig, E. Matrix Calculus, New York: Interscience, 1956.

Carver, W. B., "Systems of Linear Inequalities", Annals of Mathematics, Vol. 23, pp. 212-220, 1922.

Chow, C. K., "Boolean Functions Realizable with Single Threshold Devices" Proceedings of the IRE, Vol. 49, No. 1, pp. 370-371, January 1961.

Chow, C. K., "On the Characterization of Threshold Functions", Proceeding of the Second Annual Symposium on Switching Circuit Theory and Logic Design, pp. 34-38, September 1961.

Cline, R. E., "Representations for the Generalized Inverse of Matrices with Applications to Linear Programming", Ph.D. Dissertation, Purdue University, 1963.

Coates, C. L. and V. Supornpaibul, "On Maximum Realization of Linearly Separable Functions", Proceedings of the Sixth Annual Symposium on Switching Circuit Theory and Logic Design, pp. 12-24, October 1965.

Decell, H. P., Jr., "An Application of the Cayley-Hamilton Theorem to Generalized Matrix Inversion", SIAM Review, Vol. 7, No. 4, pp. 526-528, October 1965.

Dertouzos, M. L., "An Approach to Single Threshold Element Synthesis", IEEE Transactions on Electronic Computers, Vol.EC-13, No.5, pp.519-528, October 1964.

Dines, L. L. and N. C. McCoy, "On Linear Inequalities", Transactions of the Royal Society of Canada, Vol. 27, pp. 37-70, 1933.

Duda, R. O. and H. Fossum, "Pattern Classification by Iteratively Determined Linear and Piecewise Linear Discriminant Functions", IEEE Transactions on Electronic Computers, Vol. EC-15, No. 2, pp. 220-232, April 1966.

Forejt, D. A., "The Method of Arbitrary Choice for Training Linear Perceptrons", Ph.D. Dissertation, University of Pittsburgh, 1967.

Fu, K. S. and W. G. Wee, "An Adaptive Procedure for Multiclass Pattern Classification", Proceedings of the First Annual Princeton Conference on Information Science and System, pp. 90-94, 1967.

Fukunaga, K. and T. Ito, "An Iterative Realization of Pattern Recognition Networks," Proceedings of the NEC, Vol. 22, pp. 863-868, 1966.

Gaston, C. A., "A Simple Test for Linear Separability", IEEE Transactions on Electronic Computers, Vol. EC-12, No. 2, pp. 134-135, April 1963.

Greville, T. N. E., "Note on the Generalized Inverse of a Matrix Product", Mathematics Research Center, United States Army, The University of Wisconsin, MRCTR No. 623, January 1966.

Groner, G. F., "Statistical Analysis of Adaptive Linear Classifiers", Stanford Electronics Laboratory, TR-6761-1, SU-SEL-64-026, April 1966.

Highleyman, W. H., "A Note on Linear Separation", IRE Transactions on Electronic Computers, Vol. EC-10, No. 4, pp. 777-778, December, 1961.

Highleyman, W. H., "Linear Decision Functions with Application to Pattern Recognition", Proceedings of the IRE, Vol. 50, No. 6, pp. 1501-1514, June 1962.

Ho, Y. C., A. Arcese, R. L. Kashyap, and C. C. Blaydon, "Experiments with a Pattern Classification Technique", Cruft Laboratory TR No. 472, Harvard University, May 1965.

Ho, Y. C. and P. B. Brentani, "On Computing Optimal Control with Inequality Constraints", SIAM Journal on Control, Vol. 1, No. 3, pp. 319-348, 1963.

Ho, Y. C. and R. L. Kashyap, "A Class of Iterative Procedures for Linear Inequalities", SIAM Journal on Control, Vol. 4, No. 1, pp. 112-115, February 1966.

Koford, J. S., "Adaptive Pattern Dichotomization", Stanford Electronics Laboratory, TR-6201-1, SU-SEL-64-048, April, 1964.

Li, C. C., T. Teng, M. Zobrak, and T. W. Sze, "On Recognition of Handwriting Chinese Characters", Proceedings of the First Princeton Conference on Information Science and Systems, pp. 235-239, 1967.

Marcus, M. and H. Ming. A Survey of Matrix Theory and Matrix Inequalities, Boston: Allyn and Bacon, Inc., 1964.

Minnick, R. C., "Linear Input Logic", IRE Transactions, Vol. EC-10, No. 1, pp. 6-16, March 1961.

Motzkin, T. S. and I. J. Schoenberg, "The Relaxation Method for Linear Inequalities", Canadian Journal of Mathematics, Vol. 6, pp. 393-404, 1954.

Pratt, F. Secret and Urgent. Garden City, New Jersey: Blue Ribbon Books, 1939.

Price, C. M., "The Matrix Pseudoinverse and Minimal Variance Estimates", SIAM Review, Vol. 6, No. 2, pp. 115-120, April 1964.

Singleton, R. C., "A Test for Linear Separability as Applied to Self Organizing Machines", Self Organizing Systems, Yovits, Jacobic, and Goldstein, editors, Washington, D.C.: Sparton Books, 1962.

Smith, F. W., "Contractor Control by Adaptive Pattern Recogniton Techniques", Stanford Electronics Laboratory, TR-6762-2, SU-SEL-64-042, April 1964.

Specht, D. F., "Generation of Polynomial Discriminant Functions for Pattern Recognition", IEEE Transactions on Electronic Computers, Vol. EC-16, No. 3, pp. 308-319, June 1967.

Sze, T. W. and M. Zobrak, "A Method of Recognition of Hand Drawn Line Patterns", Proceedings of the First Annual Princeton Conference on Information Science and Systems, pp. 240-244, 1967.

Yau, S. S. and C. C. Yang, "Pattern Recognition by Using an Associative Memory", IEEE Transactions on Electronic Computers, Vol. EC-15, No. 6, pp. 944-947, December 1966.